

The Design of a Multipoint Line Topology for a Communication Network Using Genetic Algorithms*

Faris N. Abuali
Dale A. Schoenefeld
Roger L. Wainwright

Department of Mathematical and Computer Sciences
The University of Tulsa
600 South College Avenue
Tulsa, Oklahoma 74104-3189

abuali@euler.mcs.utulsa.edu
dschoen@euler.mcs.utulsa.edu
rogerw@penguin.mcs.utulsa.edu

ABSTRACT

In this paper we investigate genetic algorithms (GA) as a heuristic technique for obtaining a near optimal solution for the multipoint line topology (MLT) problem. The MLT problem involves determining minimum cost links for a collection of terminal sites that communicate with a central site in a communication network. The problem is to determine a minimum cost tree, rooted at the central site and spanning all terminal sites. Each subtree rooted at the central site corresponds to a multipoint line interconnecting the central site with a subset of the terminal sites. The constraint is that the aggregate capacity requirement of the terminals in each multipoint line not exceed the capacity of the corresponding multipoint line. The problem is more generally known as the Constrained Minimum Spanning Tree (CMST) problem. This problem is NP-complete. This research has applications to centralized information network design and can significantly reduce cost for communication network installations. The Esau-Williams algorithm is widely used to obtain a near optimal minimum cost

spanning tree that satisfies the constraint. The Esau-Williams algorithm is patterned after Kruskal's algorithm for finding an unconstrained minimum spanning tree in an undirected, connected graph and is generally considered the algorithm of choice. This research compared the Esau-Williams algorithm and several different GA representations and crossover strategies. The terminal locations were obtained from standard data sets and Euclidian distance was used as the cost. A randomly generated capacity requirement was associated with each terminal. Our GA implementation of the MLT problem outperformed the Esau-Williams algorithm in all of our test cases.

1. INTRODUCTION

A centralized data network is a network where all communication is to and from a single site called the central site or, simply, the center. It is assumed that the other sites, called terminal sites, have relatively simple equipment that cannot make routing decisions. Some terminals may be connected directly to the central site. The remaining terminals are partitioned into groups to take advantage of the economy of scale for cost versus capacity. The terminals in each group interconnect via a multipoint line and share a subtree rooted at the center. This leads to a tree topology

* Research supported by OCAST Grants AR2-004 and ARO-038 and Sun Microsystems Inc.

where there is only one possible path to the central site and, hence, only one possible path between any pair of nodes [14].

The multipoint line topology problem (MLT) is concerned with the selection of links to interconnect terminals and to connect interconnected terminals to the central site. Specifically, we are given N terminal sites which are identified by subscripts 1 through N , and a central site denoted by the subscript 0. The capacity required for traffic from site i to the central site is denoted by w_i . Symmetrically, the capacity reserved for traffic from the central site to site i is also denoted by w_i , which is assumed to be at no additional cost. We are also given c_{ij} , $0 \leq i < j \leq N$, which is the cost of connecting site i and site j with a segment of a multipoint line. The objective is to identify a collection of multipoint lines (subtrees rooted at the central site) having minimum total cost and satisfying the constraint that the aggregate capacity requirement (aggregate weight) of the terminals on any such multipoint line does not exceed the total capacity (total weight) of the multipoint line. We assume the total capacity of each multipoint line is fixed at W . This problem is known as the Constrained Minimum Spanning Tree (CMST) problem. That is, it is the classical minimum spanning tree problem (MST) with an additional constraint on the size of each subtree rooted at the central site [14]. This problem has been shown to be NP-complete [8,9,13].

In Figure 1, we show an example collection of $N = 14$ terminal sites with a central site. The capacity required for traffic from site i to the central site, w_i , is an integer in the range from 1 to 6. This is shown in Figure 1(c). Although displayed logically in a fashion that does not depict the Cartesian coordinates of the sites, the cost of connecting site i to site j , c_{ij} , with a segment of a multipoint line is assumed to be Euclidian distance. This is shown in Figure 1(b). A possible collection of multipoint lines linking terminal sites with each other and to the central site is shown in Figure 1(a). The collection of five multipoint lines represents a spanning tree of all sites, and partitions the collection of terminal sites. These values are shown in Figure 1(a). In this example, the capacity of each multipoint line is assumed to be 12. The total

cost of the indicated collection of multipoint lines is 341. The CMST problem is to find the collection of multipoint lines having the least cost subject to the constraint that the capacity of each multipoint line is not violated.

Formally, the problem can be stated as [14]:

Find $z = \min \sum_{i,j} c_{ij}x_{ij}$, $0 \leq i < j \leq N$,
subject to:

$$\sum_{i,j \in T_k} w_i x_{ij} \leq W \quad \forall k, \quad (1)$$

$$\sum_{i,j \in S} x_{ij} < |S| \quad \forall S, \quad (2)$$

$$\sum_{i,j} x_{ij} = N, \quad (3)$$

where x_{ij} is 1 if (i,j) is included in the topology and x_{ij} is 0 otherwise. Therefore, z is the cost of all of the links in the network. T_k is the k th multipoint line (subtree), defined as the set of links connecting a set of nodes to the center. k takes on values from 1 to N and T_k can be empty for some k . The first constraint, from Equation 1 above, guarantees that the total weight of the terminals on each multipoint line does not exceed the limit. The second constraint, from Equation 2, guarantees that the links chosen for the topology do not cause any cycles. The third constraint, from Equation 3, guarantees that enough links will be selected to connect the network of $N + 1$ nodes.

Kruskal's algorithm provides a greedy strategy for solving the classical MST problem. Given a connected graph with weighted edges, Kruskal's algorithm continues to select edges in order of smallest weight and accepts an edge if it does not cause a cycle. It can be shown that the algorithm does, indeed, produce a minimum spanning tree. Kruskal's algorithm forms the basis for most CMST algorithms currently used in practice. For the CMST problem, one must similarly verify that the addition of an edge does not cause a cycle. Furthermore, there is a capacity constraint on each multipoint line given by the aggregate capacity of the terminals on the multipoint line. Since the CMST problem is NP complete this variation of Kruskal's algorithm does not necessarily produce the

optimal minimum cost spanning tree for the CMST problem. One of the ways in which the algorithm can fail to produce an optimal solution is that it can leave nodes which are far from the center stranded if all of their neighbors are absorbed into subtrees that fill up.

The Esau-Williams [7] algorithm for the CMST problem improves the generic variation of Kruskal's algorithm. The Esau-Williams algorithm can be thought of as starting with each node connected directly to the center. The algorithm exchanges links to the center for links of lower cost in greedy fashion while not violating the capacity constraint. The algorithm stops when no cost effective exchanges can be made.

2. GENETIC ALGORITHMS

Genetic algorithms differ from traditional algorithms in several ways. Genetic algorithm works with a coding of the parameter rather than the actual parameter. Instead of a single point, the GA works from a population of strings. The genetic algorithm uses probabilistic transition rules, not deterministic rules, and the applications of the genetic operators causes information from the previous generation to be carried over to the next. Furthermore, genetic algorithms produce "close" to optimal results in a "reasonable" amount of time, and they make no assumptions about the problem space. In addition, genetic algorithms are fairly simple to develop and they are suitable for parallel processing.

In a genetic algorithm the parameters of the model to be optimized are encoded into a finite length string, called a chromosome. The fitness of a chromosome determines its ability to survive and reproduce offspring. The genetic algorithm creates an initial population of solutions, and then recombines them in a such way to guide the search to the most promising areas of the state space. The transition rules that produce one population from another are called genetic recombination operators. These include Crossover, Reproduction and Mutation. Crossover provides new points in the solution space to investigate. Mutation, which occurs rarely, guarantees the entire search space has the opportunity to be searched, given

enough time. Initial population size, crossover method, evaluation function, and mutation rate are parameters which have a great impact on convergence rate as well as the quality of the solution [2].

Several researchers have investigated the benefits of solving combinatorial optimization problems using genetic algorithms [1,2,4,5,6,15,19]. Genetic algorithm packages for a single processor have been available for several years. A generational GA such as GENESIS [11], and a steady-state GA such as GENITOR [17] and LibGA [3] which offers the ability to use both a generational or a steady-state approach are three example packages that are readily available. In a steady-state GA the offspring and parents are in the same pool. Each time an offspring is generated it is placed into the pool, and the weakest chromosome is removed from the pool. In a generational GA, all of the offspring are saved in a separate pool until the pool size is reached. Then the children's pool replaces the parent's pool for the next generation. These two cases represent two extremes in pool management for genetic algorithms. It is assumed the reader is familiar with the fundamentals of genetic algorithms. Davis, Goldberg and Rawling, provide an excellent in depth study of genetic algorithms [4,5,10,16].

3. THE TEST CASES

The Esau-Williams algorithm and various genetic algorithm (GA) strategies were applied to twelve test cases representing three data variations on each of four collections of sites. The number of sites, including the center, were 15, 30, 48, and 105. In each case, one of the sites served as the central site. The site locations for the size 30 and 105 cases were obtained from "standard" data sets previously used for the Traveling Salesman Problem (TSP) [18,19]. The site locations for the size 48 cases correspond to the location of the state capitols of the 48 contiguous states. The three data variations for each of the four collections of sites were determined by randomly generating three sets of terminal site capacity requirements. In each case, terminal site capacity requirements were randomly generated integers in the range from 1 to 12. In all test cases, the cost of a link

between any two sites was assumed to be the Euclidean distance. In all test cases the capacity of any multipoint line interconnecting a subset of the terminal sites and the central site was assumed to be 12.

For each test case, the Esau-Williams algorithm was executed. The Esau-Williams algorithm determines a number of multipoint lines and the location of the multipoint lines. The collection of multipoint lines is a spanning collection of subtrees rooted at the central site. Alternately stated, the Esau-Williams algorithm determines a partition of the terminal sites, the specific links required to interconnect the terminal sites in each partition, and the link required to connect each partition to the central site.

The number of multipoint lines determined by the Esau-Williams algorithm is referred to as the *EW cardinality*. The partition of the set of terminal sites created by applying the Esau-Williams algorithm is referred to as the *EW partition*. The total cost of the links participating in the collection of multipoint lines determined by the Esau-Williams algorithm is referred to as the *EW total cost*.

4. GA IMPLEMENTATION FOR THE MLT PROBLEM

The authors are not aware of any previous genetic algorithm implementation for the MLT problem. We developed two different chromosome representations for our GA implementations. One chromosome representation is a permutation of the integers 1 through N , the number of terminal sites. We refer to the permutation encoding technique simply as the *permutation encoding*. The other chromosome representation is referred to as the *group number encoding*. The chromosome representations are presented in the next section. The PMX and Cycle crossover functions were used with the permutation encoding. The uniform and single point crossover functions were used with the group numbers encoding. Both generational and steady-state strategies for evolving populations were used.

5. PERMUTATION ENCODING

Jones and Beltramo [12] investigated partitioning problems using genetic algorithms. The goal is to partition the permutation into k groups according to some criterion. They presented several methods for encoding a partition as a permutation of N objects. One of the methods is a greedy heuristic which they attribute to L. Davis. The greedy heuristic takes the first k objects in the permutation to initialize (seed) the k groups. The remaining objects of the permutation are then added to the groups one at a time. A given object is added to the group that yields the best value for the objective function. This encoding will be referred to as the Davis encoding.

Our permutation encoding is an application of the Davis encoding method to the MLT problem. A chromosome is a permutation of the N terminal sites to be partitioned into k multipoint lines. We assume that the number of multipoint lines is equal to the number of multipoint lines determined by the Esau-Williams algorithm. That is, we let k be equal to the *EW cardinality*. For the permutation encoding, the number of multipoint lines does not change as the population of chromosomes evolves, even though an alternate number of multipoint lines might further improve the topology.

The evaluation function assumes that the first k terminal sites of a chromosome are placed into k different multipoint lines, one per multipoint line as suggested by the Davis encoding method. The remaining $N-k$ terminal sites are examined individually. As each new terminal site is selected, the cost consequence of assigning it to each multipoint line is determined. That is, for each of the k multipoint lines, the new terminal site is tentatively added to the collection of vertices represented by the multipoint line. A minimum spanning tree is determined for the resulting collection of vertices. The cost of the "new" multipoint line, (a minimum spanning tree), is compared with the cost of the previous multipoint line, which is also a minimum spanning tree with one fewer terminal site. The cost consequence of assigning the new terminal site to the multipoint line is the incremental cost between the two minimum spanning trees, one

containing the new terminal site and one that does not contain the new terminal site. After determining the cost consequence of assigning the new terminal site to each of the k multipoint lines, the new terminal site is assigned to the multipoint line for which the cost consequence is minimal. The process iterates by considering the next allele in the chromosome.

The chromosome below illustrates the permutation encoding for the $N = 14$ multipoint line configuration presented in Figure 1.

8 1 12 7 14 6 2 3 4 11 9 13 5 10

The *EW cardinality* for the given data set is determined to be five. We assume that the permutation encoding is interpreted to produce a number of multipoint lines equal to the *EW cardinality*. Hence, the chromosome is interpreted to mean that, initially, the multipoint lines $T_1, T_2, T_3, T_4,$ and T_5 each contain the central site and, in addition, contain terminal sites 8, 1, 12, 7, and 14, respectively. To determine the multipoint line that contains the next terminal site in the permutation, terminal site 6, we compute, for each i ,

$$\text{MST}(T_i \cup \{\text{terminal site } 6\}) - \text{MST}(T_i).$$

MST is an operator producing the cost of the minimum spanning tree. Terminal site 6 is placed in set T_i for the i that produces the minimum cost computation. We iterate the process for the next allele, namely terminal site 2. The process continues until it is decided how to assign the last allele in the chromosome.

6. GROUP NUMBER ENCODING

Encoding a partition of the terminal sites as a string of group numbers allowed one to use the standard uniform crossover and single point crossover functions [12]. A chromosome encoded according to this strategy that would produce the same multipoint line membership as that illustrated by Figure 1 is shown below:

Terminal Site Number (Fig. 1(a))

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Group Number Representation

(2 4 3 5 1 4 4 1 1 5 3 3 5 5)

The allele in the i th position of the chromosome simply indicates the multipoint line that is presumed to contain the i th terminal site. The merit of using the group number representation is that the number of multipoint lines is decided dynamically. The permutation representation works with a fixed number of multipoint lines. The disadvantage of the group number representation is the difficulty in generating good initial populations. In fact, some initial experiments we ran with the group number representation produced final results that were not as good as the initial population generated from the permutation representation. To improve the initial population, chromosomes for the initial pool used by the permutation encoding were transformed (to preserve the multipoint line membership interpretation) into the chromosomes for the initial pool used by the group number encoding. As the population of chromosomes evolves, the number of multipoint lines represented by different chromosomes might be different.

7. RESULTS

Table I, Table II, Table III and Table IV summarize our results for executing various algorithms for the 15, 30, 48 and 105 site cases, respectively. In each of these four Tables we show results from three data variations. The three data variations were determined by randomly generating three sets of terminal site capacity requirements. In each case, terminal site capacity requirements were randomly generated integers in the range from 1 to 12. In all test cases the capacity of any multipoint line interconnecting a subset of the terminal sites and the central site was assumed to be 12. This was described previously in Section 3.

In Table I through Table IV, results are shown for the Esau-Williams algorithm and various genetic algorithm implementations. We tested our group number chromosome representation using uniform and single

point crossover operators on both the generational and steady-state models. We also tested our permutation chromosome representation using PMX and Cycle crossover operators on both the generational and steady-state models. Best cost results are depicted in each Table by an "*". In all cases our genetic algorithms were implemented using the LibGA package [3].

In the 15 site case (Table I), since the problem size was relatively small, all of the algorithms performed the same with one exception. In data set II, all of the genetic algorithms obtained a lower cost than the Esau-Williams algorithm. In the 30 site case (Table II), the GA permutation chromosome representation yielded the overall best results, independent of crossover operator or steady-state or generational model. In fact, this algorithm obtained the best results in every case except one. In the 48 site case (Table III), the GA permutation chromosome representation yielded the best results, independent of crossover operator or steady-state or generational model. This algorithm obtained the best results in every case. In the 105 site case (Table IV), the GA permutation chromosome representation again yielded the best results, independent of crossover operator or steady-state or generational model. This algorithm obtained the best results in every case tested. The group number chromosome representation GA, however, failed to generate any feasible solutions during the entire execution. This is shown as "infeas" in Table IV.

In conclusion, the results show that our permutation chromosome representation GA yields the best results for solving the MLT problem. This representation performed very well independent of the crossover operators we tested or the population management model. The group number chromosome representation GA, performed very well in small cases, but additional work is needed for larger cases. The Esau-Williams algorithm performed poorly in all of our test cases. Our permutation chromosome representation GA outperformed the Esau-Williams algorithm in every single case tested in all four problem sizes. In fact, as the problem size increases, our GA outperforms the Esau-Williams algorithm by a wider mar-

gins. In Table III and Table IV our GA produced approximately 3.0% better solution than the Esau-Williams algorithm, which in many applications can be quite significant.

8. FUTURE WORK

There are two major issues to follow up with this research. The first issue is the use of an adaptive mutation operator that increases in frequency as the population is converging. Some of our preliminary test runs using adaptive mutation performed better than some of the results reported in Tables I through IV. The second issue we are looking at is using a modified Davis encoding scheme where one does not limit the number of multipoint lines.

ACKNOWLEDGEMENTS

This research has been partially supported by OCAST Grants AR2-004 and ARO-038. The authors wish to acknowledge the support of Sun Microsystems Inc.

REFERENCES

- [1] J.L. Blanton and R.L. Wainwright, "Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms", in S. Forrest, ed., *Genetic Algorithms: Proceedings of the fifth International Conference (GA93)*, Morgan Kaufmann, San Mateo, CA.
- [2] A.L. Corcoran and R.L. Wainwright, "A Genetic Algorithm for Packing in Three Dimensions", *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing*, 1992, pp. 1021-1030, ACM Press.
- [3] A.L. Corcoran and R.L. Wainwright, "LibGA: A User-friendly Workbench for Order-based Genetic Algorithm Research", *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing*, pp. 111-118, 1993, ACM Press.
- [4] L. Davis, ed., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publisher, 1987.
- [5] L. Davis, ed., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

- [6] K.A. De Jong and W.M. Spears, "Using Genetic Algorithms to Solve NP- Complete Problems", *Proceedings of the Third International Conference on Genetic Algorithms*, June, 1989, pp. 124-132.
- [7] I.R. Esau and K.C. Williams, "On Teleprocessing System Design. A Method for Approximating the Optimal Network", *IBM System Journal*, 5:142-147, 1966.
- [8] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
- [9] B. Gavish, "Augmented Lagrangean Based Algorithms for Centralized Network Design", *IEEE Trans. on Commun.*, COM-33:1247-1257, 1985.
- [10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [11] J. Grefenstette, GENESIS, Navy Center for Applied Research in Artificial Intelligence, Navy research Lab., Wash. D.C. 20375-5000.
- [12] D.R. Jones and M.A. Beltramo, "Solving Partitioning Problems with Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 442-449, Morgan Kaufmann, 1989.
- [13] A. Kershenbaum and R. Boorstyn, "Centralized Teleprocessing Network Design", *Networks*. 13:279-293, 1983.
- [14] A. Kershenbaum, *Telecommunications Network Design Algorithms*, McGraw-Hill, 1993.
- [15] P.P. Mutalik, L.R. Knight, J.L. Blanton and R.L. Wainwright, "Solving Combinatorial Optimization Problems Using Parallel Simulated Annealing and Parallel Genetic Algorithms", *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing*, pp. 1031-1038, 1992, ACM Press.
- [16] G. Rawling, ed., *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, 1991.
- [17] D. Whitley and J. Kauth, GENITOR: A Different Genetic Algorithm, *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, Denver, Co., 1988, pp. 118-130.
- [18] D. Whitley and T. Starkweather, "GENITOR II: A Distributed Genetic Algorithm", *Journal of Experimental and Theoretical Artificial Intelligence*, 2(1990) 189-214.
- [19] D. Whitley, T. Starkweather, and D. Fuquat, "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator", *Proceedings of the Third International Conference on Genetic Algorithms*, June, 1989.

* best cost			Data Sets		
			I	II	III
Esau-Williams total cost:			* 605	573	* 645
Esau-Williams cardinality:			9	9	10
crossover	chromosome representation	generational/ steady state			
uniform	group number	generational	* 605	* 563	* 645
uniform	group number	steady state	* 605	* 563	* 645
single point	group number	generational	* 605	* 563	* 645
single point	group number	steady state	* 605	* 563	* 645
pmx	permutation	generational	* 605	* 563	* 645
pmx	permutation	steady state	* 605	* 563	* 645
cycle	permutation	generational	* 605	* 563	* 645
cycle	permutation	steady state	* 605	* 563	* 645

Table I. 14 terminal sites, 1 central site.
The mulitpoint line capacity is $W = 12$.
The terminal site requirement is random in 1-12.

* best cost			Data Sets		
			I	II	III
Esau-Williams total cost:			* 1245	1286	1307
Esau-Williams cardinality:			12	12	12
crossover	chromosome representation	generational/ steady state			
uniform	group number	generational	* 1245	1292	1295
uniform	group number	steady state	1262	1316	1295
single point	group number	generational	1262	* 1272	1295
single point	group number	steady state	1262	1316	1295
pmx	permutation	generational	* 1245	1277	* 1289
pmx	permutation	steady state	* 1245	1277	* 1289
cycle	permutation	generational	* 1245	1277	* 1289
cycle	permutation	steady state	* 1245	1277	* 1289

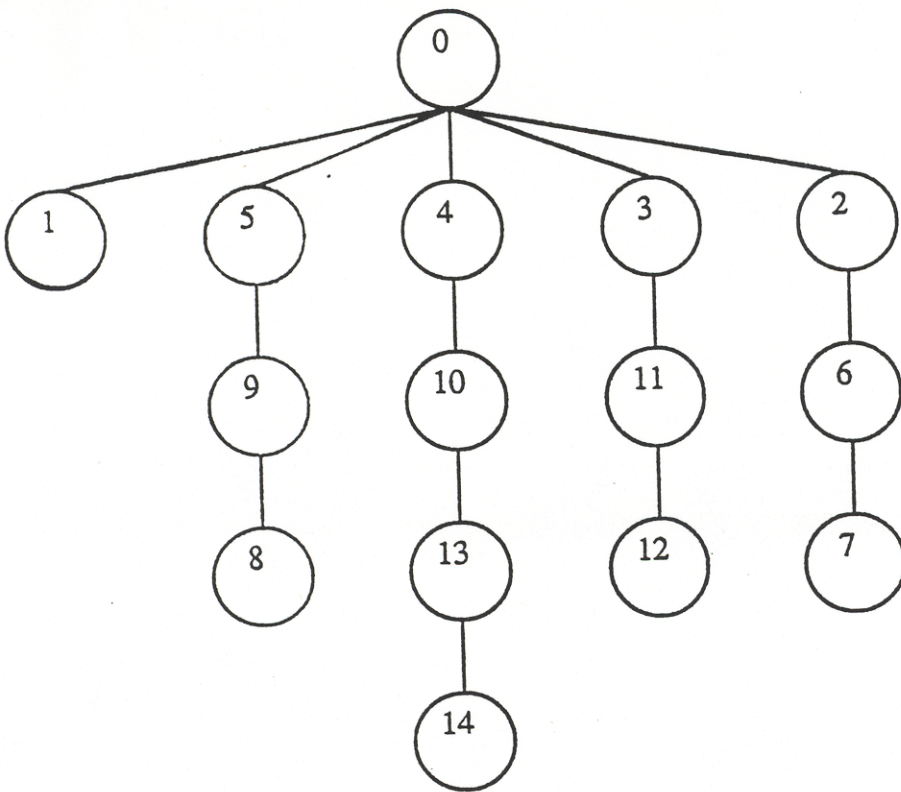
Table II. 29 terminal sites, 1 central site.
The mulitpoint line capacity is $W = 12$.
The terminal site requirement is random in 1-12.

* best cost			Data Sets		
			I	II	III
Esau-Williams total cost:			5139	5119	4664
Esau-Williams cardinality:			31	33	29
crossover	chromosome representation	generational/ steady state			
uniform	group number	generational	5018	5179	4850
uniform	group number	steady state	5018	5184	4850
single point	group number	generational	5001	5188	4662
single point	group number	steady state	5013	5285	4780
pmx	permutation	generational	* 5000	5098	* 4625
pmx	permutation	steady state	5027	* 5078	* 4625
cycle	permutation	generational	* 5000	5098	* 4625
cycle	permutation	steady state	* 5000	5098	* 4625

Table III. 47 terminal sites, 1 central site.
The multipoint line capacity is $W = 12$.
The terminal site requirement is random in 1-12.

* best cost			Data Sets		
			I	II	III
Esau-Williams total cost:			109383	105010	103696
Esau-Williams cardinality:			65	64	63
crossover	chromosome representation	generational/ steady state			
uniform	group number	generational	infeas	infeas	infeas
uniform	group number	steady state	infeas	infeas	infeas
single point	group number	generational	infeas	infeas	infeas
single point	group number	steady state	infeas	infeas	infeas
pmx	permutation	generational	106179	102989	102212
pmx	permutation	steady state	106209	103065	* 101949
cycle	permutation	generational	106360	* 102748	101971
cycle	permutation	steady state	* 106173	102832	102585

Table IV. 104 terminal sites, 1 central site.
The multipoint line capacity is $W = 12$.
The terminal site requirement is random in 1-12.



a) Multipoint Line Topology (MLT) Configuration, With partitions,
 $T_1=(0,5,8,9)$, $T_2=(0,1)$, $T_3=(0,3,11,12)$, $T_4=(0,2,6,7)$, $T_5=(0,4,10,13,14)$, with a cost of 341.

j \ i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
0		*	32	39	38	55	52	62	69	71	64	66	61	66	89	96
1			*	11	20	34	30	32	38	43	38	45	44	47	70	75
2				*	12	23	19	23	30	33	27	33	33	35	59	63
3					*	17	14	27	35	34	27	28	24	28	52	58
4						*	4	21	28	20	13	10	10	12	36	41
5							*	18	26	20	13	14	14	16	40	45
6								*	8	12	12	25	29	29	48	48
7									*	13	16	29	35	34	50	49
8										*	7	18	25	22	37	36
9											*	13	19	17	36	37
10												*	8	4	25	30
11													*	5	27	34
12														*	24	29
13															*	10

i	w _i
0	0
1	4
2	6
3	6
4	5
5	2
6	1
7	5
8	6
9	3
10	2
11	5
12	1
13	2
14	3

b) Cost matrix C_{ij} , associated with (a).

c) Weight of Terminals W_i associated with (a).

Figure 1. An example MLT Problem, with $N=14$, $K=5$, and $W=12$.