Proceedings of the 1994 ACM Symposium on Applied Computing
(SAC'94), March 6-8, 1994, Phoenix, ZA, pp. 242-246.

Designing Telecommunications Networks Using Genetic Algorithms and Probabilistic Minimum Spanning Trees*

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

Department of Mathematical and Computer Sciences
The University of Tulsa
[faris,dschoen,rogerw]@euler.mcs.utulsa.edu

Abstract

In this paper we investigate genetic algorithms (GA) as a heuristic technique for obtaining near optimal solutions to the probabilistic minimum spanning tree (PMST) problem. The PMST problem is a natural generalization of the classical minimum spanning tree (MST) problem and is frequently a more realistic model. The PMST problem addresses the circumstances that arise when not all nodes are deterministically present but, rather, nodes are present with known probabilities. Although there are some special cases that are solvable in polynomial time, it is known that the PMST problem is NP-complete.

1. Introduction

The minimum spanning tree (MST) problem is one of the classical combinatorial graph optimization problems. Finding the minimum spanning tree in a connected graph has many applications to routing and scheduling problems that include transportation and telecommunications. There are many well known polynomial time algorithms, including Prim's algorithm and Kruskal's algorithm, that can be found in almost any textbook in discrete mathematics or data structures [16].

Recently, Bertsimas defined the Probabilistic Minimum Spanning Tree (PMST) problem as a probabilistic variation of the classical MST problem [3]. The PMST problem is a natural generalization of the MST problem and is frequently a more realistic model in many of the applications where MST solutions have previously been used. Given a graph, G = (V,E), where there is a cost associated with each edge, the PMST problem assumes that each subset of nodes is active with a certain *a priori* probability. Given any spanning tree, say *T*, of G, one defines its active cost at time *a* to be the sum of the costs of the edges in a subtree T_a. T_a is the subtree of *T* obtained by retaining only those edges required to insure interconnection among the nodes that are active at time *a*. The objective in solving the PMST problem is to find the spanning tree having the smallest expected active cost.

Although there are some special cases of the problem that are solvable in polynomial time, Bertsimas showed that the PMST problem is NP complete. In fact, he showed that there are many natural and restricted cases of the problem that are also NP-complete. The objective of this research work is to apply genetic algorithm techniques to find near optimal solutions to the PMST problem for selected graphs and for selected probability density functions on the power set of all vertices. Results are compared to the expected active cost of the minimum spanning tree. To our knowledge, genetic algorithms, simulated annealing, or other techniques for obtaining near-optimal solutions have not been applied to the PMST problem.

2. The Problem

Bertsimas has formally defined the problem as follows [3]:

Given a connected graph G = (V,E), not necessarily complete, a cost function c:E->R, and a probability function p:2^V-> [0,1], the objective is to find a spanning tree, say *T*, that minimizes the expected active cost, E[L_T], where

$$E[L_T] = \sum_{S \subseteq V} p(S) L_T(S)$$

Note that the summation is taken over all subsets of V. For a spanning tree *T*, *T(S)* is the minimum subtree of *T* that is required to interconnect all the nodes in *S*. $L_{T(S)}$ is the total cost of the edges in *T(S)*.

Whenever *T* is a spanning tree of a graph and S is a subset of the vertices, we refer to *T(S)* as the *active subtree of T corresponding to S*. To illustrate *T(S)*, consider a complete graph with nine nodes and the spanning tree, *T*, indicated in Figure 1. Further suppose that, at some point in time, the collection of nodes present is *S* = {2,4,5,6,8}. Then *T(S)* is indicated in Figure 2. Note that node 1 is not in *S* and the edges (9,1) and (1,2) are absent in *T(S)*. This follows because node 9, a leaf node adjacent to node 1, is also not in *S*. Note, further, that node 7 is not in *S*, yet, edges (2,7) and (7,5) are still present in *T(S)*. This follows because there are nodes on each side (relative to *T)* of node 7 that need to be interconnected by *T(S)*.

If one assumes that node *i* is active with probability p_i and that the nodes are independent, then p:2^V->[0,1] is determined. That is, the probability associated with any subset of V is obtained by multiplying the corresponding p_i's. In this case, Bertsimas shows that the expected active cost, E[L_T], of a given spanning tree is given by the expression

$$\sum_{e \in T} c(e) \left\{ 1 - \prod_{i \in K_e} (1 - p_i) \right\} \left\{ 1 - \prod_{i \in V - K_e} (1 - p_i) \right\}$$

[*] Research supported by OCAST Grant AR2-004 and Sun Microsystems Inc.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1994 ACM 089791-647-6/ 94/ 0003 $3.50

The summation is over all edges in the tree T. Removal of any edge, e, partitions T into two subtrees. The set of vertices in one subtree is denoted by K_e and the set of vertices in the other subtree is denoted by V - K_e. The summation indicates that computing $E[L_T]$ is $O(n^2)$ where n is the number of nodes. In fact, $E[L_T]$ can be computed in O(n) time [3].

Bertsimas verified several results regarding the difficulty of the PMST problem when $p_i = p \ \forall\ i$ [3]. The problem is NP-complete if either the graph is non-complete and the costs are equal, or if the graph is complete and the costs can assume one of two distinct values, 1 or M. Further, whenever the MST is a star tree, as is the case when the graph is complete and all of the edge costs are the same, the MST is the PMST and, hence, the PMST can be found in polynomial time. In this paper we are only interested in the cases of the PMST problem that are NP-complete. Bertsimas also produced a bound on the difference between the expected active cost of the classical MST, $E[L_{MST}]$, and the expected active cost of the optimal PMST, $E[L_{TP}]$.

In instances where the probability of each node being present is very high, the problem is nearly equivalent to the classical minimum spanning tree problem and the optimal or near optimal solution represented by the minimum spanning tree is very difficult to improve upon. However, in instances where the probability of each node being present is very low, the problem is nearly equivalent to the Network Design problem, a classical NP-complete problem [3, 11].

3. Telecommunications Network Design

The PMST problem has applications in telecommunications network design. Consider a network of nodes where there are certain requirements for communication among the nodes and a desire to design an *a priori* spanning tree structure to interconnect the nodes. Further suppose that the requirement for a particular node to be "on line" at any point in time is stochastic, and that line costs are incurred only when a line is required to service a node that is "on line". Further assume that any node is able to support pass-through traffic even when it is not "on-line". The PMST represents the strategy for interconnecting the nodes that has the lowest cost over time.

There are other applications related to telecommunications network design and to VLSI design. Bertsimas also articulates an interesting example where the most cost effective organizational structure in a "spy network" is a PMST [3].

4. The Greedy Algorithm

Our greedy algorithm simply computes the classical minimum spanning tree. The selection of the tree considers edge cost but does not consider the probabilistic presence of nodes. For comparison to our genetic algorithm strategies, the expected active cost of the MST (which does reflect the probabilistic presence of nodes) is computed. On the other hand, our genetic algorithm applied to the PMST problem evaluates the suitability of many spanning trees with respect to edge cost and with respect to the probabilistic presence of nodes.

5. Genetic Algorithms

Several researchers have investigated the benefits of solving combinatorial optimization problems using genetic algorithms [1, 2, 4, 6, 10, 17]. Davis, Goldberg and Rawlins provide an excellent in depth study of genetic algorithms [8, 9, 12, 15]. It is assumed that the reader is familiar with the fundamentals of genetic algorithms (GA). The GA package used in this research is LibGA [7].

6. GA Encoding for the PMST Problem

One of the first theorems in graphical enumeration was Cayley's theorem that proved there are $n^{(n-2)}$ distinct labeled trees on n vertices [5]. Equivalently, there are $n^{(n-2)}$ distinct spanning trees of a complete graph on n vertices. Prufer provided a constructive proof of Cayley's theorem by establishing a one to one correspondence between such trees and the set of all strings of n-2 integers between 1 and n. [14] Consequently, we designed our chromosome encoding of a candidate PMST to be a string of n-2 integers between 1 and n.

For example, the chromosome corresponding to the spanning tree of a complete graph on nine vertices represented in Figure 1 is (2 7 3 2 7 2 1). The key to Prufer's construction is that any tree has at least two vertices of degree equal to one [16]. The chromosome encoding of Figure 1 is described as follows. Locate the node of degree one having the smallest label. In our case, this is node 4. Since node 2 is the node (the only node) in the tree that is adjacent to node 4, we assign 2 to the first allele in the corresponding chromosome. We continue the process on the subtree resulting from removing node 4 and the edge from node 2 to node 4. We summarize this first step and the successive steps of the construction:

2 becomes first allele, removing node 4 and edge (2,4),
7 becomes next allele, removing node 5 and edge (5,7), and
3 becomes next allele, removing node 6 and edge (3,6).
2 becomes next allele, removing node 3 and edge (2,3),
7 becomes next allele, removing node 8 and edge (7,8),
2 becomes next allele, removing node 7 and edge (2,7), and
1 becomes next allele, removing node 2 and edge (1,2).

The tree that remains consists of nodes 1 and 9 and the edge (1,9). The algorithm stops when the remaining tree has two nodes and one edge remaining.

Conversely, since the procedure is not immediately obvious, we also illustrate the strategy to construct a spanning tree in a complete graph on nine nodes from a given chromosome [16]. The key observation is that the degree of each labeled node is one more than the cardinality of the node number in the chromosome. For the chromosome (2 7 3 2 7 2 1), the degree of node 1, 2, 3, 4, 5, 6, 7, 8 and 9 is, respectively, 2, 4, 2, 1, 1, 1, 3, 1 and 1.

The smallest labeled node of degree one is node 4. Since 2 is the first allele in the chromosome, node 4 must be adjacent to node 2. After inserting an edge from node 4 to node 2, we decrement the remaining degree of node 2 from 4 to 3, and decrement the remaining degree of node 4 from 1 to 0. The process then repeats.

Several different crossover operators were used in our genetic algorithm for solving the PMST problem. The first crossover operator we considered was the standard single point crossover. To see how this works on spanning trees, consider two parent chromosomes (2 7 3 2 7 2 1) and (3 3 1 9 7 5 1) shown in Figures 1 and 3, respectively. Assume the single point crossover point is between the third and fourth alleles. This produces children chromosomes of (2 7 3 9 7 5 1) and (3 3 1 2 7 2 1) shown in Figure 4 and Figure 5, respectively. Even though there are some similarities between the parent spanning trees and the resulting children spanning trees in this simple example, we observed that the single point crossover operator often produced children that were quite dissimilar to their parents. Hence, the single point operator appears to be disruptive.

Next, we considered several asexual crossover operators (one parent producing one child). The first asexual crossover operator is a left circular shift of the chromosome by a fixed number of positions. As an example, consider the chromosome in Figure 1 after a left circular shift of four positions. The result is the spanning tree shown in Figure 6. Notice that this crossover operator preserves the degrees of all of the nodes; only the links are rearranged.

We also implemented a crossover operator that exchanges alleles in two randomly chosen positions in the chromosome. Figure 7 is the result of exchanging the third and sixth alleles in Figure 1. An interesting result occurs when two alleles are exchanged in a chromosome where both alleles appear only once. For example, Figure 8 shows the result of exchanging the third and seventh alleles of Figure 1 (allele values one and three, which each appear only once in the chromosome). This results in a simple exchange of the two nodes in the resulting spanning trees. In Figure 1, node 1 is adjacent to nodes 2 and 9, and node 3 is adjacent to nodes 2 and 6. In Figure 8 this is reversed: now node 3 is adjacent to nodes 2 and 9, and node 1 is adjacent to nodes 2 and 6. In general, we found the asexual crossover operators to be less disruptive than the single point crossover.

The mutation operator consists of randomly selecting an allele and altering its value to be a random number between one and the number of nodes. For example, consider Figure 1 where the fourth allele is altered to a value of one. The results are shown in Figure 9. Notice that, in this instance, a simple change results from Figure 1 to Figure 9, which is a desired characteristic of any mutation operator. An edge is removed connecting nodes 2 and 3, and is reattached between nodes 1 and 3. A close inspection of the difference between the chromosomes in Figure 1 and Figure 9 yields this same insight. Notice, in the chromosome of Figure 9, that the degree of node 2 decreases by 1, and the degree of node 1 increases by one in comparison to the original chromosome in Figure 1. This necessitates the single edge change described above.

One of the benefits of our chromosome encoding scheme used to represent a spanning tree, is that all possible chromosomes are feasible solutions. That is, any allele can take on any value from one to the number of nodes, and this will always result in a spanning tree. Consider one extreme where all allele values are identical, (4 4 4 4 4 4 4), for example. The reader is encouraged to verify that this represents a star topology with node 4 as the hub. At the other extreme, consider another case where each allele value is unique, for example (1 2 3 4 5 6 7). The reader can verify that this results in a tree that is a list. The links are between nodes 8-1-2-3-4-5-6-7-9.

7. The Test Cases

As indicated above, when each node is present with a probability that is near 1, the PMST problem is nearly the same as the MST problem. Hence, if nodes are present with probability nearly 1, the genetic algorithm strategy will have a difficult time evolving a candidate PMST that is better than the MST. In our research, two types of data sets were devised. In each case, a complete graph on 20 nodes was generated. The nodes were randomly placed on a 100 x 100 grid. The cost of each edge is the Euclidean distance. With the first type of data set, the probability associated with each node was 0.1. The first type of data set was replicated five times using different random number seeds to generated coordinate positions for the nodes. For each of the five repetitions of the first type of data set, the expected active cost of the classical MST was compared to the expected active cost of various spanning trees evolved by applying genetic algorithm strategies with various options for crossover and mutation.

The second type of data set was deliberately contrived so that the the classical MST was not likely to be the PMST. To make the problem interesting, the classical MST is generated before p_i, the probability associated with node *i*, is assigned. The leaf nodes of the classical MST were assigned a probability of 0.8. The internal nodes of the classical MST were assigned a probability of 0.1. The intent was that the PMST would be more likely to have internal nodes of the classical MST as external nodes and conversely. Again a genetic algorithm was executed to determine a near optimal PMST. The expected active cost of the classical MST was compared to the expected active cost of the GA evolved candidate PMST.

8. Results

Table 1 shows the results of the PMST problem for the first type of data set where there are 20 nodes and each node is active with a probability of 0.1. We tested five different crossover operators: *Uniform, Single Point, Shift Left 1, Alter Allele, and Swap 2 Alleles.* For each crossover operator, we tested three types of mutation operators: *None* (no mutation), *Alter Allele*, and *Swap 2 Alleles.* Notice that the mutation operators are also crossover operators. Each combination of crossover operator and mutation operator was run five different times with a different random placement of the 20 nodes within the region. The resulting expected active costs are denoted in Table 1. The results from the greedy algorithm are given at the bottom of Table 1 for comparison.

The "*" in Table 1 indicates every instance where the genetic algorithm obtains a better solution than the greedy algorithm. When considering the *Uniform* crossover operator, the *Alter Allele* mutation performed best. For the *Single Point* crossover, the *Alter Allele* mutation also performed best. Notice the dismal performance of the *Shift Left 1* crossover operator. For the *Alter Allele* crossover operator , both the *Alter Allele* mutation and the *Swap 2 Alleles* mutation performed well. For the *Swap 2 Alleles* chromosome operator, the *Alter Allele* mutation performed best.

Now consider the best single performance in each of the five random cases: 51.5 in case I, 61.0 in case II, 57.0 in case III, 63.6 in case IV, and 61.0 in case V. In each instance this occurred when *Alter Allele* was the mutation operator. The crossover operators yielding the single best result over the five cases included *Uniform, Alter Allele* and *Swap 2 Alleles*, but not *Single Point* or *Shift Left 1.*

The conclusion from our research with the first type of data set is that the best mutation operator for our encoding of the PMST problem is definitely *Alter Allele.* The best crossover operators, using the *Alter Allele* mutation operator, are *Uniform*, *Alter Allele*, and *Swap 2 Alleles.* Our findings indicate that *Single Point* crossover and *Shift Left 1* crossover, regardless of the mutation operator, are too disruptive for our encoding of the PMST problem and are not good candidates for the genetic algorithm. Furthermore, the GA results for the *Alter Allele* mutation operator with any of the three best crossover operators is superior to the greedy algorithm in every instance tested.

The results for the contrived data set are indicated in Figure 10 and Figure 11. The classical MST for twenty nodes is shown in Figure 10. The classical cost (length) of the MST is 265. After letting the internal nodes of the classical MST be active with probability 0.1 and the leaf nodes of the classical MST be active with probability 0.8, the expected active cost of the classical MST is 228.

The GA algorithm was then applied to the same data set as indicated in Figure 10. That is, the node locations remained the same. Further, as with the data set indicated in Figure 10, the probability associated with each node was assigned to be 0.1 or 0.8 depending upon whether the node is a leaf node or an internal node of the classical MST. The *Uniform* crossover operator and the *Alter Allele* mutation operator were used with the GA algorithm. The "near optimal" PMST evolved by the GA algorithm is indicated in Figure 11. The expected active cost of the spanning tree in Figure 11 is 210. For comparison, the classical cost (the total length of all of the edges, disregarding the probabilities) is 403. Because of the manner in which the probabilities were contrived, it is intuitive that the PMST would tend to "reverse" the internal nodes and the leaf nodes of the MST. That such characteristics are reflected in the spanning tree evolved by the GA algorithm is a credit to the GA techniques.

9. Future work

One major contribution of this research is the observation that the one to one correspondence between spanning trees and the set of all strings of n-2 integers between 1 and n [5, 14, 16] leads to a natural GA encoding of the PMST problem. A second major contribution is the identification of crossover operators and mutation operators that promote the evolution of good near optimal solutions to the PMST problem.

Future research should verify that the results are scalable to complete graphs with more nodes. Other efforts should show that the techniques are still useful in case the graph is not complete and, hence, there are infeasible chromosomes. In cases where there are a small number of nodes and the computation is possible, more work is needed to compare the GA solution to the optimal PMST. Finally, more effort is needed to identify additional applications where the desired optimization is approximated by finding a probabilistic minimum spanning tree.

Acknowledgements

This research has been supported by OCAST Grant AR2-004 and by Sun Microsystems Inc.

References:

[1] Abuali, F. N., Schoenefeld, D. A. and Wainwright, R. L. "The Design of a Multipoint Line Topology for a Communication Network Using Genetic Algorithms", to appear, *Proceedings of the Seventh Oklahoma Conference on Artificial Intelligence*, November, 1993.

[2] Abuali, F. N., Schoenefeld, D. A. and Wainwright, R. L., "Terminal Assignment in a Communication Network Using Genetic Algorithms", to appear, *Proceedings of the Twenty Second Annual Computer Science Conference*, March, 1994

[3] Bertsimas, D. J., "The Probabilistic Minimum Spanning Tree Problem," *Networks*, Volume 20: 245-275, 1990.

[4] Blanton, J. L. and Wainwright, R. L. "Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms", *Proceedings of the Fifth International Conference on Genetic Algorithms* (ICGA-93), Stephanie Forrest, Editor, Morgan Kaufmann Publisher, 1993, pp. 452-459.

[5] Cayley, A., "A Theorem on trees," *Quarterly Journal of Mathematics*, 23:376-378, 1889.

[6] Corcoran, A. L. and Wainwright, R. L. "A Genetic Algorithm for Packing in Three Dimensions", *Proceedings of the 1992 ACM Symposium on Applied Computing*, March 1-3, 1992, pp. 1021-1030, ACM Press.

[7] Corcoran, A. L. and Wainwright, R. L., "LibGA: A User-Friendly Workbench for Order-Based Genetic Algorithm Research", *Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing*, February, 14-16, 1993, pp. 111-117, ACM Press.

[8] Davis, L. ed., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publisher, 1987.

[9] Davis, L. ed., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[10] De Jong, K. A. and Spears, W. M., "Using Genetic Algorithms to Solve NP-Complete Problems", *Proceedings of the Third International Conference on Genetic Algorithms*, June, 1989, pp. 124-132.

[11] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

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

[13] Kershenbaum, A., *Telecommunications Network Design Algorithms*, McGraw-Hill, 1993.

[14] Prufer, H., "Neuer beweis eines Satzes uber Permutationen," *Arch. Math. Phys.*, 27:742-744, 1918.

[15] Rawlins, G., ed., *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, 1991.

[16] Skiena, S., *Implementing Discrete Mathematics, Combinatorics and Graph Theory with Mathematica*, Addison Wesley, 1990.

[17] Whitley, D., Starkweather, T, and Fuquat, D., "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator", *Proceedings of the Third International Conference on Genetic Algorithms*, June, 1989.

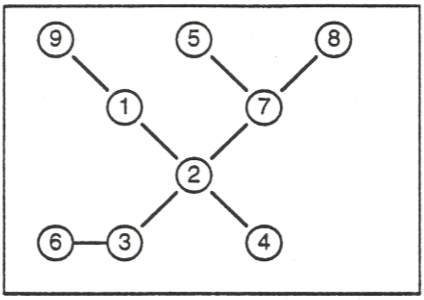

Figure 1: An example spanning tree (ST) for a complete graph on nine nodes. The chromosome is (2 7 3 2 7 2 1).

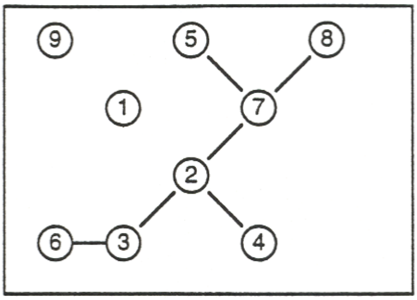

Figure 2: An active subtree of the ST in Fig 1. The subtree corresponds to S = {2,4,5,6,8}.

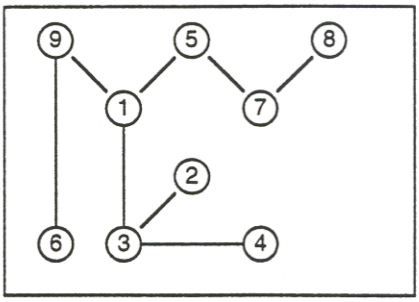

Figure 3: A second example spanning tree for a complete graph on nine nodes. The chromosome is (3 3 1 9 7 5 1).

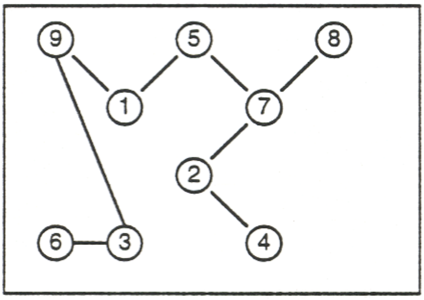

Figure 4: The first ST resulting from a single point crossover of Fig 1 and Fig 3. The chromosome is (2 7 3 9 7 5 1).

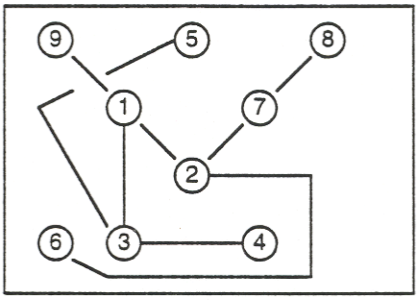

Figure 5: The second ST resulting from a single point crossover of Fig 1 and Fig 3. The chromosome is (3 3 1 2 7 2 1).

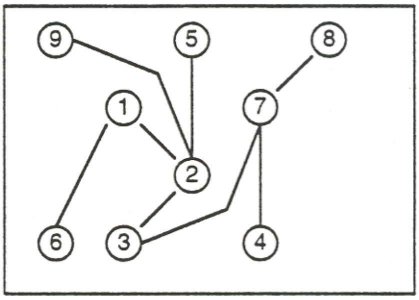

Figure 6: The ST resulting from a left circular shift of four positions in Fig 1. The chromosome is (7 2 1 2 7 3 2).

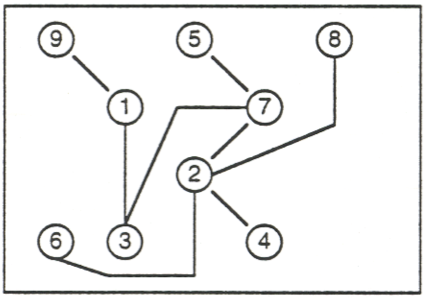

Figure 7: The ST resulting from exchanging the third and sixth alleles in Fig 1. The chromosome is (2 7 2 2 7 3 1).

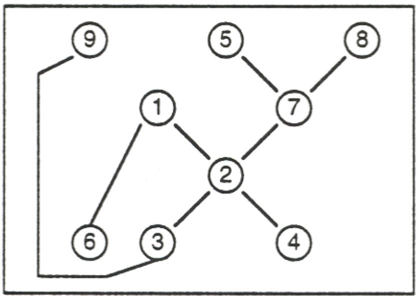

Figure 8: The ST resulting from exchanging the third and seventh alleles in Fig 1. The chromosome is (2 7 1 2 7 2 3).

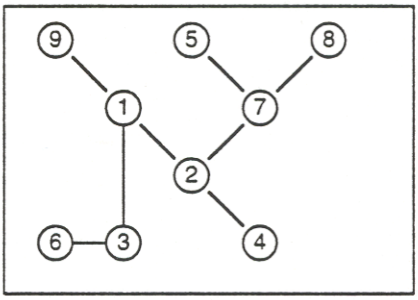

Figure 9: The ST resulting from a mutation on Fig 1. The chromosome is (2 7 3 1 7 2 1).

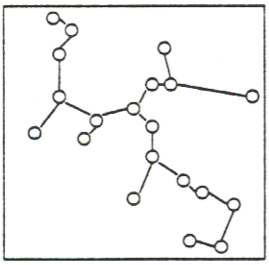

Figure 10: The classical MST for the contrived data. Expected cost: 228.

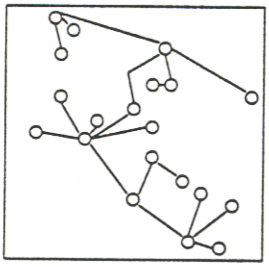

Figure 11: The GA near optimal PMST for contrived data. Expected cost: 210.

Algorithm		Expected Active Cost				
Genetic Algorithm		Test Cases				
Crossover	Mutation	I	II	III	IV	V
Uniform	None	54.9	63.8*	60.3*	70.8*	72.5
	Alter Allele	53.3*	61.0*	57.5*	63.6*	61.5*
	Swap 2 Alleles	54.5	60.7*	59.8*	64.8*	63.6*
Single Point	None	61.1	70.6	73.6	77.6*	74.2
	Alter Allele	52.3*	61.1*	60.3*	73.4*	64.3*
	Swap 2 Alleles	57.5	61.9*	65.9	71.5*	67.5
Shift Left 1	None	110.9	113.5	113.6	127.9	111.0
	Alter Allele	73.4	74.9	85.2	103.7	95.1
	Swap 2 Alleles	104.0	91.9	91.3	107.5	87.2
Alter Allele	None	54.9	61.2*	57.2	70.9*	62.3*
	Alter Allele	52.5*	61.1*	57.0*	69.5*	61.0*
	Swap 2 Alleles	51.5*	62.6*	56.3*	70.1*	60.3*
Swap 2 Alleles	None	75.9	79.1	63.7	88.2	71.1
	Alter Allele	51.5*	61.9*	60.3*	64.0*	63.2*
	Swap 2 Alleles	67.3	75.7	64.4	79.1*	73.4
Greedy		54.1	67.6	62.6	87.1	66.9

Table 1: PMST problem, 20 nodes, type one data set.