



MULTICAST TREE GENERATION USING TABU SEARCH

Muhammad Atif Tahir¹, Habib Youssef², A. Almulhem³ and Sadiq M. Sait⁴

1: Lecturer, Department of Information and Computer Science, KFUPM, Dhahran, Saudi Arabia

2: Associate Professor, Departement des sciences de informatique, Faculte des Sciences de Monastir, Tunasia.

3: Assistant Professor, Department of Computer Engineering, KFUPM, Dhahran, Saudi Arabia

4: Professor, Department of Computer Engineering, KFUPM, Dhahran, Saudi Arabia

Correspondence address: P.O. Box 1132, KFUPM, Saudi Arabia, Fax: 9663 860-2174,

E-mail: atif@ccse.kfupm.edu.sa

ABSTRACT

The research and industrial communities have recently been striving to make the Internet capable of supporting multicast traffic sources. A multicast tree for a given multicast source is a tree rooted at the source and all its leaves being members in the multicast group. Tree cost is measured by the utilization of tree links. The problem to be solved is to find a minimum cost multicast tree. This problem has been proven to be NP-Complete. In this paper, we present a Tabu Search algorithm to construct a minimum-cost multicast tree. The proposed algorithm is compared with other existing multicast algorithms. Results show that on almost all test cases, Tabu Search algorithm exhibits more intelligent search of the solution subspace and is able to find better solutions than other reported multicast algorithms.

Keywords: Multicast Tree, Tabu Search, Optimization, Shortest Path.

(NP_COMPLETE)

1. INTRODUCTION

The Internet is widely recognized to deliver connectivity to the data world. The monetary investment in the Internet is enormous; however it operates as a unicast communication network. There is an emerging trend for group communication also called multicasting. This has a potential application in the business world where it will help increase the ability of organizations to communicate and collaborate, leveraging more value from network investment. Examples are the transmission of corporate messages to employees, video and audio conferencing for remote meetings and teleconferencing, transmission over networks of live TV or radio news and entertainment programs, and many more. These new applications are compelling the need for advances in traffic handling to overcome bottlenecks [Kosiur, 1998]. Figures 1 and 2 show the difference between unicast and multicast data flows [Kosiur, 1998].

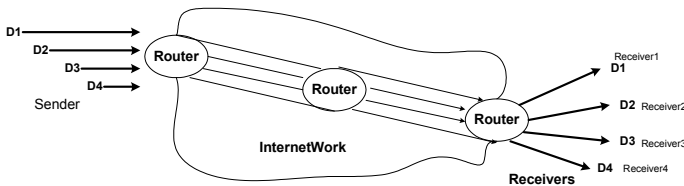


Fig. 1 Unicast Dataflow

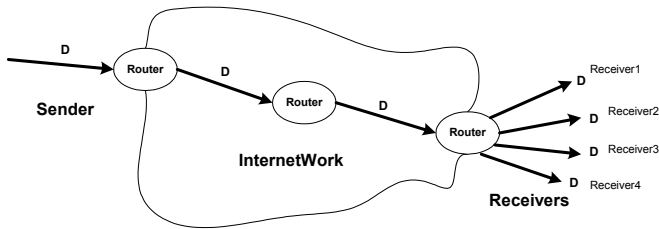


Fig. 2 Multicast Dataflow

To support multicast communication efficiently, one of the key issues that needs to be addressed is routing. Routing primarily refers to the determination of a set of paths to be used for carrying the messages from a source node to all destination nodes. It is important that the routes used for such communications consume a minimal amount of resources. In order to use network resources as little as possible while meeting the network service requirements, the highly recommended solution involves the generation of a multicast tree spanning the source and destination nodes. Minimum Steiner tree (MST) algorithms attempt to minimize the total

cost of the multicast tree. Total cost of the multicast tree is generally defined as the sum of costs of all edges in the multicast tree. The cost is usually measured as the bandwidth consumed by the tree. The minimum Steiner problem is known to be NP-Complete [Waxman, 1988.].

The paper is organized as follows. Section 2 defines the multicast routing problem. Section 3 surveys some recently proposed heuristics. Section 4 describes our Tabu Search based multicast tree design. Simulation results and comparison with other reported heuristics are presented in Section 5. Section 6 concludes the paper.

2. PROBLEM FORMULATION

The communication network is modeled as a graph $G=(V,E)$ where V is a set of nodes and E is a set of edges (links). A link represents a physical connection between two nodes. The cost of an edge e is denoted by $cost(e)$ where $cost(e)$ is positive real number. A directed network model is assumed, i.e., the links $e=(u,v)$ and $e=(v,u)$ are used interchangeably. The link cost is a function of the amount of traffic traversing the link e and the expected buffer space needed for that traffic [3, 4]. The minimum Steiner tree problem [Waxman, 1988.] is to determine a multicast tree connecting the source node s to every destination node $d \in D$, such that the total cost of this tree is minimal. Mathematically, the problem is to find a tree $T = (V_T, E_T)$ where $V_T \subseteq V$ and $E_T \subseteq E$ such that the total cost of this tree $\sum_{e \in E_T} cost(e)$ is minimized subject to the constraint $\{s\} \cup D \subseteq V_T$

3. LITERATURE REVIEW

The objective of the minimum Steiner tree problem is to minimize the total cost of the multicast tree. Very few algorithms have been proposed for the minimum Steiner tree problem in directed network [Hwang and Richards, 1992]. The minimum Steiner tree problem is known to be NP-complete. KMB [Waxman, 1988.] is an efficient minimum Steiner tree heuristic for undirected networks. The KMB heuristic uses Prim's minimum spanning tree algorithm [Prim, 1957.] during its computation. Prim's algorithm is optimal only for symmetric networks. Thus the cost performance of the KMB heuristic may be optimized if it is applied to asymmetric networks. The worst-case time complexity of the KMB heuristic is $O(MV^2)$, where M is the size of the multicast group and V is the number of nodes in the network. The total cost of trees generated using KMB heuristic in symmetric networks is on the average 5% worse than the cost of the optimal minimum Steiner tree [Doar and Leslie, 1993.]. RPM [Dalal and Metcalfe, 1978.] and [Deering and D. Cheriton, 1990.] is another Steiner tree heuristic for undirected networks. RPM is used in practice because it requires only limited information to be available at each node in order to construct a reverse shortest path multicast tree.

An efficient solution for the Steiner tree problem with application to multicast routing is presented by Ricudis [Ricudis, et.al 1999.]. The solution is based on a hybridized Genetic algorithm with a hill climbing technique that facilitates better local exploration of the solution search space.

4. TABU_BASED MULTICAST TREE DESIGN

The number of possible multicast trees in a computer network even of medium size is extremely large. Further because of the constrained nature of the problem and the various cost parameters, it is not clear what constitutes the best tree. Modern iterative heuristics such as tabu search have been found effective in tackling this category of problems, which have an exponential and noisy search space with numerous local optima [Sadiq and Youssef, 1999.]. These iterative algorithms are heuristic search methods, which perform a non-deterministic but intelligent walk through the search space. In this paper, we present a Tabu Search algorithm to find a low cost multicast tree.

4.1. Overview of Tabu Search

Tabu Search (TS) was introduced by Fred Glover [Glover, 1990.] and [Glover, 1996.] as a general iterative metaheuristic for solving combinatorial optimization problems. Tabu Search is conceptually simple and elegant. It is a form of local neighborhood search. Each solution $S \in \Omega$ has an associated set of neighbors $N(S) \subseteq \Omega$ where Ω is the set of feasible solutions. A solution $S' \in N(S)$ can be reached from S by an operation called a move to S' . TS moves from a solution to its best admissible neighbor, even if this causes the objective function to deteriorate. To avoid cycling, solutions that were recently explored are declared forbidden or tabu for a number of iterations. The tabu status of a solution is overridden when certain criteria (aspiration criteria) are satisfied. The Tabu Search algorithm is given in Fig 3.

4.2. Proposed Tabu Search Based Algorithm

Our Algorithm assumes that sufficient global information is available to the source, i.e., the source node of the network has complete information about all network links to construct a multicast tree.

Initial Solution: The algorithm starts with an initial feasible solution $S \in \Omega$ built in a greedy fashion as follows. A minimum cost Steiner tree is constructed using Dijkstra's shortest-path algorithm [Kosiur, 1998] starting from the source. This results in a set of superpaths. A superpath is defined as a set of edges whose starting and ending nodes are the root of the tree and any node in $S \cup D$. We call this tree as sink tree of source s . For example, consider a network as shown in Fig 4(a), here $s = \{A\}$ and $D = \{B, C, D, F\}$. The sink tree for source s is shown in Fig (b). The encoding solution is path based, where a solution is encoded as an array of k elements where each element is a superpath representing a branch of the multicast tree and $k = |D|$ i.e., cardinality of set D , where D are the members of the multicast group.

Ω : Set of feasible solution
 S : Current Solution
 S^* : Best admissible solution
 $Cost$: Objective function
 $N(S)$: Neighborhood of solution S
 V^* : Sample of neighborhood solutions
 T : Tabu list
 AL : Aspiration Level

Begin

1. Start with an initial feasible solution $S \in \Omega$.
 2. Initialize tabu lists and aspiration level.
 3. For fixed number of iterations Do
 4. Generate neighbor solutions $V^* \subset N(S)$.
 5. Find best $S^* \in V^*$.
 6. If move S to S^* is not in T Then
 7. Accept move and update best solution.
 8. Update tabu list and aspiration level.
 9. Increment iteration number.
 10. Else
 11. If $Cost(S^*) < AL$ Then
 12. Accept move and update best solution.
 13. Update tabu list and aspiration level.
 14. Increment iteration number.
 15. End If
 16. End If
 17. End For
- End**

Fig. 3. Algorithmic description of a short-term Tabu Search (TS).

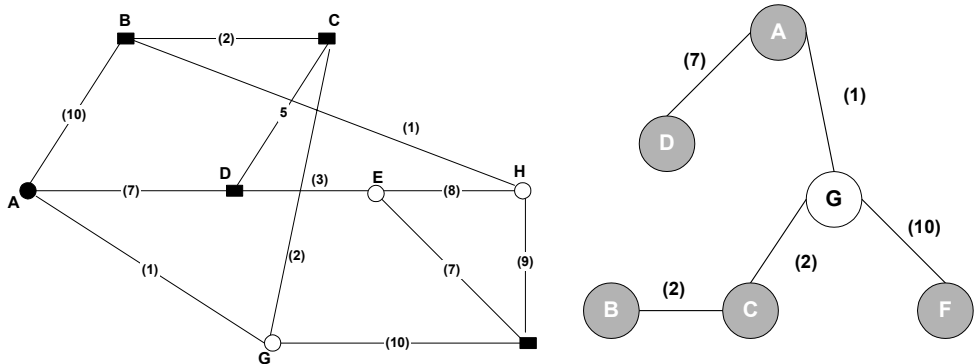


Fig. 4. (a) The example network (b) Sink Tree for source A, Cost = 22.

The encoding for the initial solution corresponding to the sink tree of Fig 4 (b) is as follows

| | | | | |
|------|---------|---------|------------|-----------|
| 0 | 1 | 2 | 3 | |
| A, D | A, G, F | A, G, C | A, G, C, B | Cost = 22 |

Neighborhood Solutions:

For generating neighbors, we choose a neighborhood structure using “delete and add” operations. To reduce the size of the solution space to be searched by Tabu, we construct a sink tree for each destination using Dijkstra's Shortest Path Algorithm. Fig 5 shows the sink trees generated for the network of Fig 4 (a). Each iteration begins by generating a set V^* of neighboring solutions. In our algorithm, the set V^* is dynamic, i.e., it varies from one iteration to another. At each iteration, we randomly delete one superpath from the encoding of current solution and then generate different feasible solutions by adding superpaths from one of the destinations sink trees as shown in Fig 6.

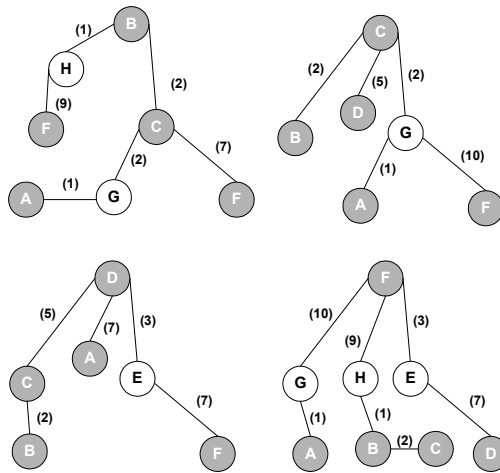


Fig. 5. Sink Trees for destinations $D=\{B, C, D, F\}$

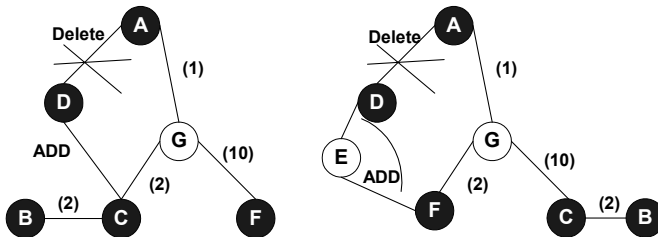


Fig. 6. Two possible Neighbors from current solution.

Among the neighbors, the one with the best cost is selected, and considered as new current solution for the next iteration. For example, the two new trees of Fig 6 are shown in Fig 7 and are encoded as follows

| | 0 | 1 | 2 | 3 | |
|------|---------|---------|---------|------------|-----------|
| 7(a) | D, C | A, G, F | A, G, C | A, G, C, B | Cost = 20 |
| 7(b) | D, E, F | A, G, F | A, G, C | A, G, C, B | Cost = 25 |

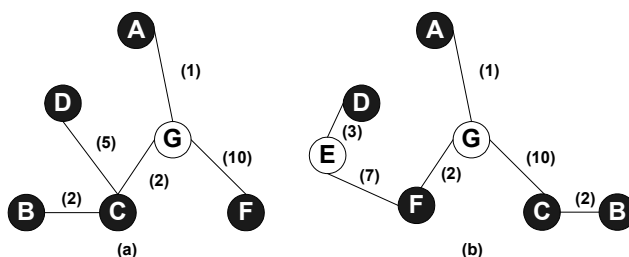


Fig. 7. Final solutions from current solution

The new multicast tree of Fig 7(a) is selected for next iteration and considered as new current solution. As mentioned above, we randomly delete one superpath from the current solution in the next iteration. Thus, for all candidate solutions in the current iteration, it is guaranteed that at least one solution will give a multicast tree.

Tabu Moves:

If a superpath is deleted at iteration *i*, then reintroducing the same superpath in an add operation is tabu. The tabu list size is set to 7.

Aspiration Criterion:

As is common in tabu search, a tabu status of a move can be overridden if accepting it results in a better cost. This means in this case a tabu superpath will be inserted anyhow.

Termination Rule:

We have used a fixed number of iterations as a stopping criterion. We experimented with different values of iterations, and found that for all the test cases, the tabu search algorithm converges within a maximum of 500 iterations.

5. SIMULATION RESULTS

The tabu search algorithm described in this paper has been tested on several randomly generated networks. A random generator [Salama, et. al, 1997.] (based on Waxman's generator [Waxman, 1988.] with some modifications) was used to create these networks. A multicast routing simulator MCRSIM [H. F. S. et al, 1995] developed at North Carolina State University was used to generate random graphs as described in [Salama, et. al, 1997.]. In this simulator n nodes are randomly distributed over a rectangular coordinate grid. Each node is placed at a location with integer coordinates. The Euclidean metric is then used to determine the distance between each pair of nodes. On the other hand, edges are introduced between pairs of nodes u and v with a probability that depends on the distance between them.

Many existing Steiner tree heuristics including KMB [Waxman, 1988.], RPM [Dalal and Metcalfe, 1978.], [Deering and D. Cheriton, 1990.] have been implemented in MCRSIM. For each run of the experiment, a random set of links to interconnect the fixed nodes is generated. Random background traffic for each link is also generated. A random source node and a multicast group of randomly chosen destination nodes are selected. The equivalent bandwidth of each link's background traffic was a random variable uniformly distributed between minimum background traffic B_{\min} and maximum background traffic B_{\max} . As the range of the link loads, i.e., the difference between B_{\min} and B_{\max} , increased, the asymmetry of the link loads also increased, because the load on the forward link $e = (u, v)$ is independent of the load on the backward link $e' = (v, u)$. The tabu search based algorithm was run on 20, 30, 40, 50, 60, 70, 80, 90, and 100 node random graph with $\alpha = 0.15$ and $\beta = 2.2$. The B_{\min} and B_{\max} are assumed to be *10 Mbps* and *150 Mbps* respectively. The default link capacity is *155 Mbps*. Following sets of experiments are carried out.

In order to analyze the performance of the proposed tabu search algorithm, we compare it with KMB as well as with RPM. There are three sets of experiments. In the first set of experiments, tabu search is compared with KMB and RPM for symmetric networks. In the second set of experiments, tabu search is compared for asymmetric networks, and in the third set of experiment, a completely unloaded network is taken and kept adding multicast (MC) sessions and constructing the corresponding MC trees until the cumulative tree failure rate exceeded 15%. A MC session consisted of a random source node generating VBR video traffic with an equivalent bandwidth of 0.5 Mbps, and a MC group of randomly chosen destination nodes. The experiment was repeated with different MC groups

Figures 8 and 9 show the cost comparison between RPM, KMB and Tabu under fixed group size of 10 and different networks for Symmetric Networks and Asymmetric Networks respectively. Figures 10 and 11 show the cost comparison between RPM, KMB and Tabu different group size for symmetric networks and asymmetric networks respectively. Our proposed tabu search based heuristic was able to identify better trees for all Test Networks. Figure 12 shows the number of multicast sessions versus cost for a network of 20 nodes and group size of 5 between Tabu and KMP. The number of multicast sessions is the same for

Tabu and KMP heuristics, and both algorithms are able to identify alternate paths for the saturated links. Tabu was even able to identify a better tree in terms of cost in almost all cases for different multicast sessions as compared to KMP. Figure 13 shows the number of multicast sessions versus cost between Tabu and RPM for a network of 20 nodes and group size of 5. The sharp line in the graph at multicast session 52 for RPM indicates the failure of RPM to identify the alternate paths when the links got saturated.

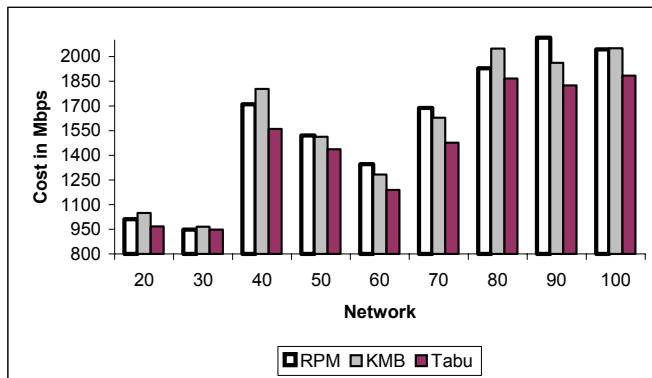


Fig. 8. Symmetric load. Cost comparison of RPM, KMB and Tabu. Group size =10.

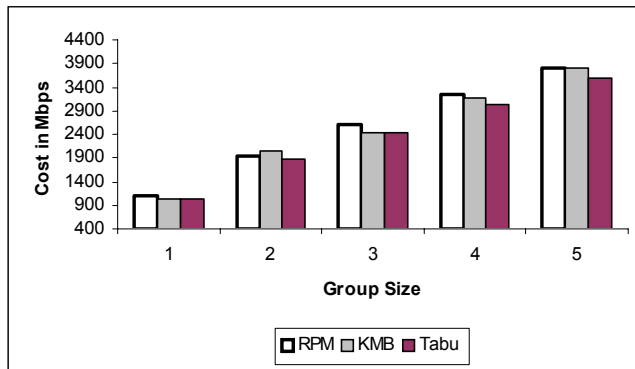


Fig. 9. Symmetric load. Cost comparison of RPM, KMB and Tabu. Network size =80.

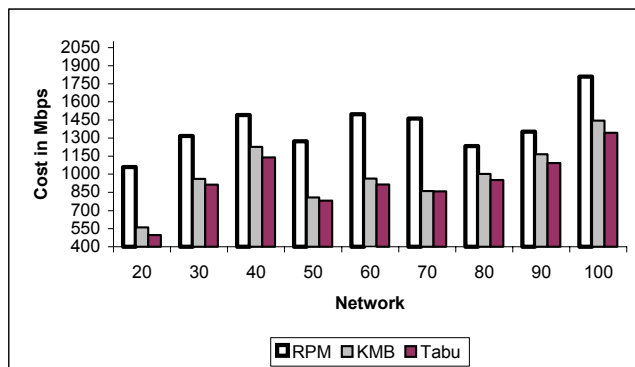


Fig. 10. Asymmetric load. Cost comparison of RPM, KMB and Tabu. Group size =10.

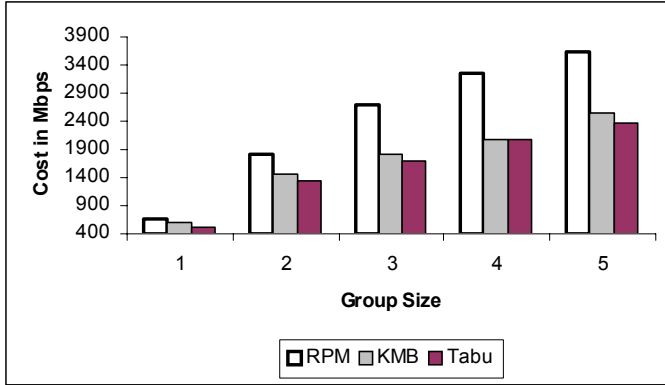


Fig. 11. Asymmetric load. Cost comparison of RPM, KMB and Tabu. Network size=100.

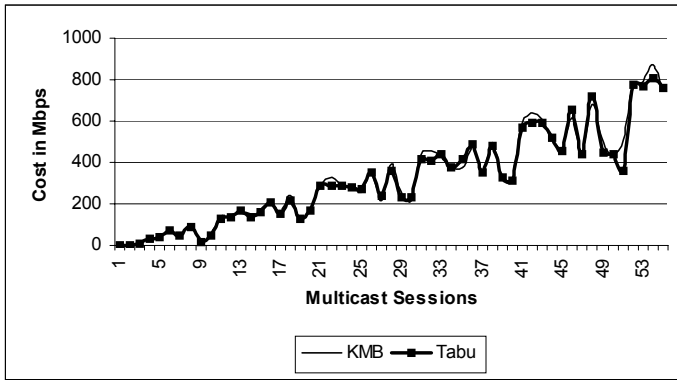


Fig. 12. Number of multicast session versus cost between KMB and Tabu. Network Size =20, Group size=5.

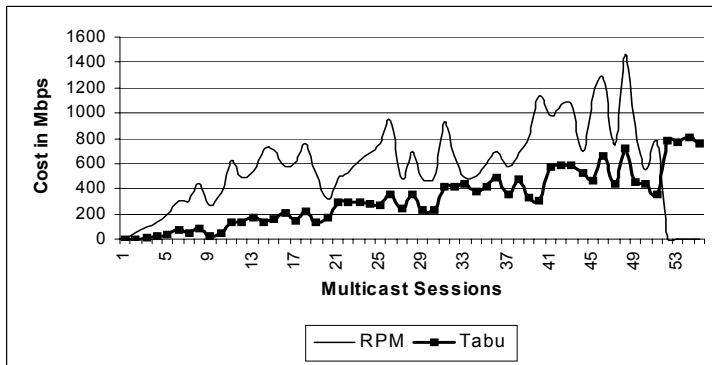


Fig. 13. Number of multicast session versus cost between RPM and Tabu. Network Size =20, Group size=5.

Fig 14 (a) shows how well focused is TS on the good solution subspace. As is clear from the figure, more than 50% of the multicast trees found and evaluated by TS were in the good solution subspace (barchart highly skewed towards the left), i.e. in the cost interval [1590-1700). Fig 14 (b) tracks with time the total number of solutions found by the proposed TS algorithm for various cost intervals. The plot clearly indicates that as more iterations are evaluated, TS keeps converging to better solution subspaces. For example, very few solutions (<20) are found beyond 300 iterations in cost interval [1700-2100). In contrast, nearly all solutions found and evaluated after the first 300 iterations are in the cost interval [1590-1700). Figures 14 (a) and 14 (b) clearly indicate that TS has been well tuned to the problem addressed in this work. Fig 14 (c) tracks the cost of the best solution over time. As is clear, for small and medium size networks (<60), TS converges within a maximum of 500 iterations.

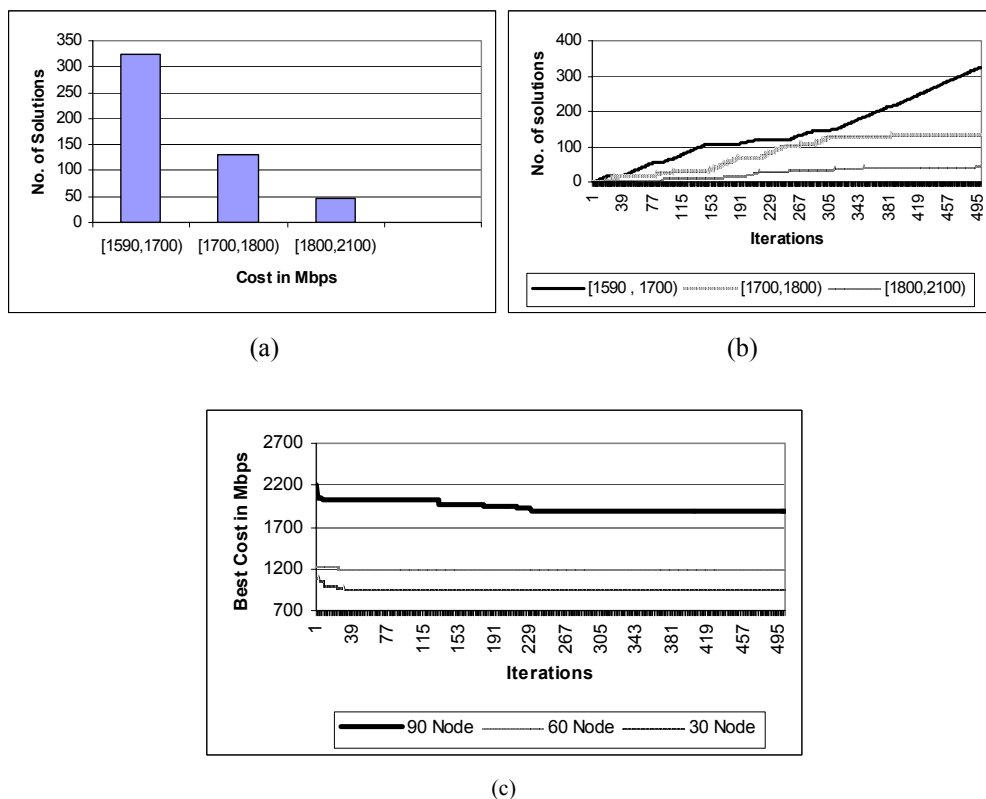


Fig. 14. Group Size = 10 and U = 0.05sec.

- (a) Total no. of Solutions evaluated by Tabu Search during 500 iterations for different cost ranges. Network size = 100 nodes.
- (b) Total no. of solutions evaluated by tabu search vs. iteration number, for different cost ranges. Network size = 100 nodes.
- (c) Cost of best solution found by tabu search vs. iteration number for 3 different networks.

The proposed algorithm has polynomial time complexity. The most expensive step of our heuristic is the initial step where Dijkstra's shortest-path algorithm is used for generating sink trees for the source and the destinations. The worst time complexity of this step is $O(MV^2)$ where M is the number of multicast members including the source and V is the number of nodes in the network. In tabu search, one iteration costs $O(M)$. Thus, for k iterations, the cost becomes $O(Mk)$. The expected time complexity of proposed TS algorithm is $O(Mk + MV^2)$. The parameter Mk is usually much smaller than MV^2 .

6. CONCLUSION

In this paper, we have presented a Tabu Search Algorithm for multicast routing problem. The proposed TS algorithm was always able to find a multicast tree if one exists. TS is better in terms of tree cost as compared to KMP and RPM. Further, as time elapsed, TS progressively zoomed towards a better solution subspace, a desirable characteristics of approximation iterative heuristics.

ACKNOWLEDGEMENTS

Authors acknowledge, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia for support.

REFERENCES

1. Dalal, Y. and Metcalfe, R.,1978, "Reverse path forwarding". *Communications of the ACM*, vol. 21, pp. 1040-1048, December.
2. Deering, S. and Cheriton, D.,1990, "Multicast routing in datagram internet works and extended lans". *ACM Transactions on Computer Systems*, vol. 8, pp. 85-110, May.
3. Doar, M. and Leslie, I.,1993, "How bad is naive multicast routing". *In Proceedings of IEEE INFOCOM 93*, pp. 82-89.
4. Glover, F., 1996, "Tabu search and adaptive memory programming- advances, applications and challenges". *Tech. rep.*, College of Business, University of Colorado at Boulder.
5. Glover, F.,1990, "Tabu search; a tutorial". *Tech. rep.*, University of Colorado, Boulder, February .
6. H. F. S. et al,1995, "MCRSIM simulator source code and Users Manual". *Center for Advanced Computing and Communication*, North Carolina State University, Raleigh, Available by anonymous ftp at <ftp.csc.ncsu.edu/pub/rtcomm>.
7. Hwang, F. and Richards, D.,1992, "Steiner Tree Problems". *Networks*, vol. 22, pp. 55-89, January.
8. Kosiur, D., 1998, "IP Multicasting: The Complete Guide to Interactive Corporate Networks". Willey Computer Publishing.

9. Prim, R.,1957, "Shortest connection networks and some generalizations". *The Bell Systems Technical Journal*, vol. 36, pp. 1389-1401, November.
10. Rampal, S. and Reeves, D.,1995, "An evaluation of routing and admission control algorithms for multimedia traffic". *Computer Communications*, vol. 18, pp. 755-768, October.
11. Ricudis, C. S., Saltouros, M., and Angelopoulos, M.,1999, "An efficient evolutionary algorithm for (near-) optimal steiner trees calculation: An approach to routing of multipoint connections". *Third International Conference on Computational Intelligence and Multimedia Applications*, pp. 448-453, Sept .
12. Sait, Sadiq, M., and Youssef, H.,1999,"General iterative algorithms for combinatorial optimization". *IEEE Computer Society*.
13. Salama, H. F., Reeves, D. S., and Viniotis, Y.,1997, "Evaluation of multicast routing algorithms for real-time communication on high speed networks". *IEEE Journal on Selected Areas in Communication*, vol. 15, pp. 332-346, April 1997.
14. Waxman, B. M., 1988, "Routing of multipoint connections". *IEEE Journal on Selected Areas in Comm.*, vol. 6, pp. 1617-1622, Dec.