

# A Lagrangean Heuristic For The Degree Constrained Minimal Spanning Tree Problem

Rakesh Kawatra, Minnesota State University, USA

## ABSTRACT

*In this paper we present a new heuristic procedure to solve the degree constrained minimal spanning tree problem. This procedure uses Lagrangian relaxation of the integer programming formulation of the problem to get a lower bound for the optimal objective function value. A subgradient optimization method is used to find multipliers that give good lower bounds. A branch exchange procedure is used after each iteration of the subgradient optimization to generate a feasible solution from an infeasible Lagrangean solution. Computational results are given for problems with up to 300 nodes. The heuristic procedure presented here gives optimal solutions in most instances. For problem sets that were not solved optimally, the gap between the lower bound and the feasible solution was less than  $10^{-2}$  percent.*

**Keywords:** Degree-constrained trees, heuristic, Lagrangean relaxation

## 1. INTRODUCTION

The Degree Constrained Minimal Spanning Tree (DCMST) problem is frequently encountered in the design of communication and computer networks. It consists of finding transmission links to connect a set of geographically remote terminal sites to a central node. The number of ports available on a terminal node restricts the number of links incident on it. Garey and Johnson [5] have shown the DCMST problem to be NP-hard by reducing it to an equivalent symmetric Traveling Salesman Problem. Several methods for solving this difficult problem have been developed. A Lagrangian-based branch and bound algorithm was used by Volgenant [13] to find the optimal solution of this problem with up to 150 nodes. Gabow [4] used a branch exchange algorithm to find an approximate solution to the problem. Gavish [7] used a branch and bound heuristic to solve the problem with up to 200 nodes. Narula and Ho [10] also used a branch and bound heuristic to solve the DCMST problem with up to 100 nodes, while Yamamoto [14] developed an algorithm based upon finding the minimum common basis of two matroids. Other heuristic approaches suggested for solving this problem include a genetic algorithm by Zhou and Gen [15], a heuristic algorithm by Bolden, Deo, and Kumar [1] that was implemented on parallel processors, and a branch & bound method based on Lagrangean relaxation used by Savelsbergh and Volgenant [12] to solve problems with up to 150 nodes.

In this paper, we present a Lagrangean relaxation based heuristic to find a good feasible solution of the DCMST problem. We relax the DCMST problem using Lagrangean relaxation method. The solution to the relaxed problem gives a lower bound to the optimal solution value of the DCMST problem. This technique to get a lower bound of the DCMST problem has been used very successfully in [7], [12], and [13]. These lower bounds were used in the branch & bound procedures to get good heuristic solution to the DCMST problem. In this paper, we use a very different approach than used by [7], [12], and [13]. We use subgradient optimization method to get Lagrange multipliers that give good lower bounds. We use a new branch exchange procedure after each iteration of the subgradient procedure to get a feasible solution from the infeasible Lagrangean solution. The best lower bound and the best feasible solutions are retained when the subgradient method terminates. The lower bound value is used to estimate the quality of the solution given by the branch exchange procedure. We found that the gaps between the feasible solution and the lower bound using our algorithm are tighter than reported previously. Additionally, we have provided results for much larger networks.

The outline of the paper is as follows: In Section 2, we present the integer-programming formulation of the DCMST problem. A Lagrangean relaxation of the problem and a subgradient optimization method to find a lower bound of the problem is described in Section 3. A branch exchange procedure embedded in the subgradient optimization method is also described in Section 3. The results of the experiment are presented in Section 4. Finally, some concluding remarks are presented in section 5.

**MODEL FORMULATION**

We use the following notation in the model:

- $C_{ij}$  = annual cost of the link directly connecting node  $i$  to node  $j$ ;
- $S = [2,3,\dots,N]$  is a set of terminal nodes to be connected to the central node (node 1);
- $r_j$  = limit on the maximum number of links that can end at a terminal node  $j$ ;

The decision variables are:

- $X_{ij} = 1$ , if there is a directed link from node  $i$  to node  $j$ ; 0 otherwise;
- $Y_{ij}$  = flow on the link directed from node  $i$  to node  $j$ ;

The integer programming formulation of the problem is given below.

$$Z_{IP} = \min \left\{ \sum_{i=2}^N \sum_{j=1}^N C_{ij} X_{ij} \right\} \tag{1}$$

subject to:

$$\sum_{j=1}^N X_{ij} = 1 \quad i = 2,3,\dots,N \tag{2}$$

$$\sum_{j=1}^N Y_{ij} - \sum_{j=2}^N Y_{ji} = 1 \quad i = 2,3,\dots,N \tag{3}$$

$$Y_{ij} \leq (N-1)X_{ij} \quad i = 2,3,\dots,N, j = 1,2,\dots,N \tag{4}$$

$$\sum_{i=2}^N X_{ij} \leq r_j \quad j = 2,3,\dots,N \tag{5}$$

$$\sum_{i=2}^N \sum_{j=1}^N X_{ij} = (N-1) \tag{6}$$

$$Y_{ij} \geq 0 \quad \forall i, j \tag{7}$$

$$X_{ij} = 0 \text{ or } 1 \quad \forall i, j \tag{8}$$

In the above model, constraints (2) guarantee that there is exactly one link originating from each terminal. Constraints (3) are flow conservation constraints, where  $Y_{ij}$  is interpreted as the flow on the directed link from node  $i$  to node  $j$ . The central node has an aggregate demand of  $(N-1)$  units of a commodity, with a supply of 1 unit from each terminal node. Constraints (4) ensures that there would be traffic flowing directly from node  $i$  to node  $j$  only if there is a directed link from node  $i$  to node  $j$ . Constraints (5) are the degree constraints, which restrict the number of links terminating at each terminal node to a predetermined number. Constraint (6) guarantees that a total of  $(N-1)$  links will be installed in the network. Constraints (2), (3), (5)-(8) ensure that the  $X_{ij}$  variables define a spanning tree rooted at node 1.

**SOLUTION PROCEDURES**

In this study we use a Lagrangian relaxation approach to generate lower bounds for the DCMST problem. This approach has been used very successfully to obtain tight lower bounds for this problem in [7], [12], and [13]. For an application-oriented survey of Lagrangean relaxation, see Fisher [3].

We form a relaxation of problem  $Z_{IP}$  by multiplying constraint set (5) with a vector of non-negative Lagrangian multipliers  $\mu = (\mu_2, \mu_3, \dots, \mu_N)$  and add it to the objective function. This results in the following problem.

$$L(\mu) = \min \left\{ \sum_{i=2}^N \sum_{j=1}^N \tilde{C}_{ij} X_{ij} - \sum_{j=2}^N \mu_j r_j \right\} \tag{9}$$

subject to (2)-(4), (6)-(8),

where  $\tilde{C}_{ij} = C_{ij} + \mu_j \qquad \qquad \qquad \forall i, j$

**Procedure for Evaluating  $L(\mu)$**

The function  $L(\mu)$  is evaluated by solving a minimal spanning tree problem rooted at the central node, which for a given vector of Lagrange multipliers  $\mu$  can be accomplished very easily using Prim's algorithm [11].

We know that for a given vector of Lagrange multipliers  $\mu \geq 0$ ,  $L(\mu)$  is a lower bound to  $Z_{IP}$ . However, we are interested in finding the tightest bound, which can be calculated, by finding

$$L(\mu^*) = \max_{\mu} \{L(\mu)\}.$$

Computing  $\mu^*$  is very difficult; however, approximate values of these vectors can be found by using a subgradient optimization method [9]. In this method, we begin with an initial vector of multipliers  $\mu^0$ , and at iteration  $p$  adjust the multipliers using the following rule:

$$\mu_j^{p+1} = \mu_j^p + s_p V_j^p \qquad \qquad \qquad \forall j = 2, 3, \dots, N$$

Where

$$V_j^p = \sum_{i=2}^N X_{ij} - r_j \qquad \qquad \qquad \forall j = 2, 3, \dots, N$$

and  $s_p = \frac{\alpha(Z^* - L(\mu_p))}{\|V_j^p\|^2}$ .

In the above equation  $\| \cdot \|^2$  denotes the Euclidean norm,  $Z^*$  is the best available overestimate of the optimal solution value, and  $\alpha$  is a scalar multiplier which satisfies the condition  $0 < \alpha \leq 2$ . The value of  $\alpha$  is initially set to 2 and is reduced during the course of the search. This method for adjusting the Lagrange multipliers has been used very successfully in [6-8]. The subgradient optimization procedure is terminated when the total number of iterations exceed a prespecified number or when a feasible solution is found with  $Z_{IP} = L(\mu^*)$  or when the gap between the lower bound and the best upper bound is less than a specified value.

**A Branch Exchange Procedure**

In this section we describe a branch exchange procedure, which is used after each iteration of the subgradient optimization procedure to generate a feasible solution to  $Z_{IP}$  from the infeasible Lagrangean solution. The best feasible solution is retained when the subgradient optimization algorithm is terminated.

For this procedure we define the following additional variables:

$link_{ij}$  = a directed link from node  $i$  to node  $j$ ; if  $link_{ij}$  is added to the network, then  $X_{ij} = 1$ ;  
 $f_i$  = termination node of the link directed away from node  $i$ , i.e., if  $X_{ij} = 1$ , then  $f_i = j$ ;

$$G^+ = \left\{ i : \sum_{j=2}^N X_{ji} > r_i \right\}, \text{ i.e., the set of terminal nodes for which the degree constraint is violated;}$$

$$G^- = \left\{ i : \sum_{j=2}^N X_{ji} < r_i \right\}, \text{ i.e., the set of terminal nodes for which the degree constraint is slack;}$$

$$R_{ij} = \tilde{C}_{i_{f_i}} - \tilde{C}_{ij}, \text{ i.e., the change in the total cost if } link_{ij} \text{ replaces } link_{i_{f_i}}$$

**Procedure A**

Step 1. If  $G^+ = \phi$ , then STOP.

Step 2. Compute, for each  $i \in G^+$  and  $j \in G^-, R_{ij}$ .

Step 3. Compute  $(i^*, j^*) = \text{Arg min}_{i,j} \{R_{ij}\}$ .

Step 4. Remove  $link_{i_{f_i}}^*$  from the network and replace it with  $link_{i_j}^*$ . Go to Step 1.

**NUMERICAL RESULTS**

The effectiveness of the heuristic procedure was investigated by performing computational tasks on a series of test problems for  $N = 40$  to  $N = 300$ . The data for the computational experiments for number of nodes in the network from  $N=40$  to  $N=100$  was taken from the CRD data set which is available from OR-Library [1]. There are 10 CRD problems for each  $N = 40, 70, 100$ , thus giving 30 CRD problems in all. We selected the Euclidean distances between these points as the entries in the cost matrix. For networks with number of nodes  $N=200$  and  $N=300$  we took the data from the SYM data set available from Craig etal. [2]. For each  $N=200$  and  $N=300$ , there are 10 SYM problems, which gives us a total of 20 SYM problems. The limit on the number of links incident on a terminal node was varied from 2 to 4. The overestimate of the objective function value was updated after each iteration of the subgradient optimization algorithm and set equal to the best feasible solution value found till that point. After each iteration an upper bound  $\bar{Z}$  and a lower bound  $\underline{Z}$  (the best-known heuristic solution and the

best-known Lagrangean relaxation, respectively) are maintained. The initial value of scalar  $\alpha$  was set to 2, and halved whenever  $L(\mu)$  did not improve in 20 successive iterations.

The search for the optimal Lagrangean multipliers was terminated if the total number of iterations exceeded 90 or if  $\bar{Z}$  changed by less than 0.004 in 30 successive iterations or when the gap between  $\bar{Z}$  and  $\underline{Z}$  was less than 0.01 percent of the value of  $\bar{Z}$ . The subgradient optimization and the branch exchange procedures were coded in FORTRAN 77 and the experiments were performed on a VAX-4500 computer. The computational results are summarized in Table 1. In the table the lower bound and the heuristic solution reported for each set of Number of nodes and r parameter values represents the average of the lower bound and heuristic solution for the 10 problems solved for the set.

**Table 1: Computation Results**

Number Of nodes	r	Lower bound	Heuristic solution	Gap	CPU time (seconds)
40	2	4215	4215	0.000%	1
40	3	4214	4214	0.000%	
40	4	4214	4214	0.000%	
70	2	5634	5634	0.000%	3
70	3	5627	5627	0.000%	
70	4	5627	5627	0.000%	
100	2	6713	6714	0.007%	8
100	3	6710	6710	0.000%	
100	4	6710	6710	0.000%	
200	2	9564	9565	0.009%	184
200	3	9556	9556	0.000%	
200	4	9556	9556	0.000%	
300	2	11534	11535	0.008%	684
300	3	11522	11522	0.000%	
300	4	11522	11522	0.000%	

The computational results indicate that the heuristic solution finds optimal solutions most of the time. For problem sets that could not be solved optimally, the gap between the heuristic solution and the lower bound is less than  $10^{-2}$  percent. Our heuristic solution takes very little CPU time. However, the CPU time required by the heuristic grows exponentially with the number of nodes in the network.

## CONCLUSIONS

In this paper we presented a Lagrangean relaxation based heuristic to solve the degree constrained minimal spanning tree problem. We used a Lagrangean relaxation method to find a lower bound of the optimal solution value of the DCMST problem. Subgradient optimization method was used to find good estimate of the Lagrange multipliers that provide the best lower bound. A new branch exchange procedure was used after each iteration of the Lagrangean relaxation to generate a feasible solution from an infeasible Lagrangian solution. The best lower bound given by the Lagrangian relaxation method is used to estimate the quality of the best solution given by the branch exchange procedure. Results from our tests on networks with up to 300 nodes indicate that our heuristic solves the problem optimally most of the time. For non-optimal solutions, the gap between the feasible solution and the lower bound is less than  $10^{-2}$  percent.

**AUTHOR INFORMATION**

**Rakesh Kawatra** is a professor of information systems at Minnesota State University, Mankato. He has a BTech. from IIT, Delhi, MBA from IIM, Bangalore, and PhD from The University of Iowa. His work has been published in European Journal of Operational Research, Computers and Operations Research, and other journals. His research focuses on topological design of communication networks.

**REFERENCES**

1. Boldon, B., Deo, N., and Kumar, N., "Minimum weight degree constrained spanning tree problem: Heuristics and implementation on an SIMD parallel machine," *Parallel Computing*, vol. 22, 369-382, 1996.
2. Craig, G., Krishnamoorthy, M., and Palaniswami, M., "Comparison of heuristic algorithms for the degree constrained minimum spanning tree," *Metaheuristics: Theory and Applications*, I.H. Osman and J.P. Kelly, Eds. Kluwer, Boston, 1996.
3. Fisher, M.L., "The Lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 27, 1-18, 1981.
4. Gabow, H.N., "A good algorithm for smallest spanning trees with a degree constraint," *Networks*, vol. 8, 201-208, 1978.
5. Garey, M.R., and Johnson, D.S., *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
6. Gavish, B., "Formulations and algorithms for the capacitated minimal directed tree problem," *Journal of the ACM*, vol. 30, 118-132, 1983.
7. Gavish, B., "Topological design of centralized computer networks-formulations and algorithms," *Networks*, vol.12, 355-377, 1982.
8. Gavish, B., "Augmented Lagrangian Based Algorithm for Centralized Network Design," *IEEE Transaction on Communications*, vol. 33, 1247-1257, 1985.
9. Held, M., Wolfe, P., and Crowder, H.D., "Validation of Subgradient Optimization," *Mathematical Programming*, vol. 6, 62-88, 1974.
10. Narula, S.C., and Ho, C.A., "Degree constrained minimum spanning tree," *Computers and Operations Research*, vol. 7, 239-249, 1980.
11. Prim, R.C., "Shortest Connection Networks and Some Generalizations," *Bell Systems Technical Journal*, vol. 36, 1389-1401, 1957.
12. Savelsbergh, M., and Volgenant, T., "Edge exchanges in the degree constrained minimum spanning tree problem," *Computers and Operations Research*, vol. 12, 341-348, 1985.
13. Volgenant, A., "A Lagrangean approach to the degree-constrained minimum spanning tree problem," *European Journal of Operational Research*, vol. 39, 325-331, 1989.
14. Yamamoto, Y., "The held-karp algorithm and degree-constrained minimum 1-trees," *Mathematical Programming*, vol. 15, 228-231, 1978.
15. Zhou G., and Gen, M., "Approach to the degree-constrained minimum spanning tree problem using genetic algorithms," *Engineering Design and Automation*, vol. 3, 156-165, 1997.