Models for the Degree Constrained Minimum Spanning Tree Problem with Node-Degree dependent Costs

Luís Gouveia, *CIO-DEIO, Faculdade de Ciências da Universidade de Lisboa, Portugal*
Pedro Moura, *CIO-DEIO, Faculdade de Ciências da Universidade de Lisboa, Portugal*

Keywords: Spanning Tree, Linear Relaxation, Valid Inequalities, Knapsack Problem

1. Introduction

Consider an undirected graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ and let $c_e \geq 0$ denote the cost of edge $e \in E$. The problem of finding a *minimum spanning tree* in G is well known and is known to be solved in polynomial time (see, for instance, [10] and [13]). In general, the problem becomes *NP-hard* when additional constraints on the spanning tree topology are required. One example is given by the well studied *Degree constrained Minimum Spanning Tree* which is known to be *NP-hard* (see the recent work of Cunha and Lucena, [5], and the references inside). This constraint is usually motivated by the need to impose a limit on the number of ports in each node.
In the problem studied in this paper, we also consider a more general objective function involving costs that are associated to multiplexing equipment to be installed on the nodes and that depend on the degree of the nodes. Furthermore the multiplexing equipment is available in *modules* (each module permits the connection of a given fixed number of links). The degree dependent cost of a node i is given by the *cost of the base equipment* (which is considered in any non-leaf node since only in this case, one requires to verify whether that node is the destination of an incoming message or the message should be sent to another node along one of the remaining links connected at that node) plus a *cost per module* (dependent on the number of modules installed at each non-leaf node), $f^k = P_1 + k \cdot P_2$, $\forall k = 1, \ldots, K$ where P_1 and P_2 are respectively, the *cost of the base equipment* and *cost per module* and k is the number of modules needed to support all the rerouting connections at node i. We let B denote the module capacity and let $D + 1$ denote the maximum number of links that a node can support. Then the maximum number of modules that need to be installed at node i is $K = \lceil \frac{D+1}{B} \rceil$.

Let $d(i)$ denote the degree of node $i \in V$ in the optimal solution. Based on the previous observations, we define an objective function that besides a term on the sum of the edge costs, $\sum_{e \in E} c_e x_e$, also involves an extra term on the node degrees, $\sum_{i \in V} f(d(i))$ such that:

$$f(d(i)) = \begin{cases} 0 & \text{if } d(i) = 1 \\ f^1 & \text{if } 2 \leq d(i) \leq B \\ f^k & \text{if } (k-1) \cdot B + 1 \leq d(i) \leq k \cdot B \qquad 2 \leq k \leq K \end{cases} \tag{1}$$

The cost function $f(d(i))$ is typical of the so-called *Network Loading Problems* (see for instance [3], [4] and [11]). In these problems, the *loading* constraints are usually imposed on the arcs of the network. As far as we know, Belotti *et al.* [1] is the first work to study and discuss models for a network design problem involving discrete costs associated to facilities installed at the nodes. Our work discusses a similar type of problem where the topology of the network is also required to be a tree.
In section 2 we present several integer programming formulations for the problem. The linear programming relaxations of the several models are also related. Section 3 presents our preliminary computational experiments.

2. Integer Programming Formulations

In this section we present several integer programming formulations for the problem. We start by presenting a generic non-linear integer programming formulation and then we proceed by presenting two integer linear programming formulations.

Let $E(i)$ denote the set of all edges adjacent to node i and let ST denote the convex hull of incidence vectors of spanning trees. Consider binary variables $x_e = 1/0$ indicating whether edge e is in the solution or not, as well as integer variables u_i such that $u_i = d(i) - 1$ in the solution. The problem can be modeled as an integer non-linear program (NLF) (see Fig. 1).

$$(NLF)\ min \quad \sum_{e \in E} c_e\, x_e + \sum_{i \in V} f(u_i + 1) \tag{2}$$

$$s.to: \quad \{x_e, e \in E\} \in ST \tag{3}$$

$$\sum_{e \in E(i)} x_e = 1 + u_i \qquad \forall\, i \in V \tag{4}$$

$$u_i \leq D \qquad \forall\, i \in V \tag{5}$$

$$x_e \in \{0, 1\} \qquad \forall\, e \in E \tag{6}$$

$$u_i \in \mathbb{N}_0 \qquad \forall\, i \in V \tag{7}$$

Figure 1: Non-linear Formulation

Constraints (4) are definitional and relate the two sets of variables. Constraints (5) are upper bound degree constraints. Constraint (3) is given in a generic form and indicates that the set of edges e such that $x_e = 1$ is a *spanning tree*. It can be represented by several equivalent sets of linear constraints (see for instance [12]). These sets either involve only the x_e variables (leading, in general, to exponential sized sets of inequalities) or involve other variables as well (leading, in general, to compact formulations).

Next we present two integer linear programming formulations for the problem. The first formulation is a 0-1 discretization and uses 0-1 variables indicating the degree of a given node. The second formulation has a stronger linear programming relaxation bound and is based on the fact that we can view the feasible set of the second model as the intersection of two integer polyhedra.

The first linear programming formulation, a discretized model, uses a variable for each node and each possible value of the degree of the node (provided it is greater than 1). Using a suitable variable exchange relation, we can apply it directly to the original non-linear model in order to obtain an integer linear programming model. This technique has already been applied to other problems and, as far as we know, has been first suggested by Gouveia [7] for the *Capacitated Minimum Spanning Tree Problem.* Two recent examples of the technique together with advantages of using the discretized model are given by Gouveia and Saldanha da Gama [8] in the context of *Concentrator Location Problems* and by Correia, Gouveia and Saldanha da Gama [2] for the *Variable Sized Bin Packing Problem.* Although this technique has an obvious disadvantage, namely the creation of many discretized binary variables, the experiments reported in the previous works suggest that the information attached to the discretized variables usually permit us to generate new sets of strong valid inequalities which lead to models with a much stronger linear programming relaxation. The reader should also be referred to the recent paper by Uchoa *et al.* [14] on the *Capacitated Minimum Spanning Tree Problem* where the authors make the most of this discretization technique by combining the use of non-trivial new valid inequalities with sophisticated pricing techniques. Another advantage is the obvious fact that more general cost functions can be modeled by using the discretized variables.

The problem studied here exemplifies these two situations namely, using the discretized variables to model more general cost functions and to generate new valid inequalities. In the context of our problem, the discretized model uses "new" binary variables $y_i^d, (i \in V,\ d \in \{1, \ldots, D\})$ indicating whether node i has degree equal to $d + 1$ in the tree solution. One interesting remark about applying the discretization technique to our problem is that here, due to the degree constraint on the nodes, we have much fewer discretized variables than in the other previously referred works. We obtain the discretized model from the previous non-linear model (see Fig. 1) by using the following set of linking constraints:

$$u_i = \sum_{d=1}^{D} d \cdot y_i^d \tag{8}$$

that permit us to replace old variables with the new variables.

The advantage of this model is obvious since the term $f\left(\sum_{d=1}^{D} d \cdot y_i^d + 1\right)$ in the objective function (2) can be simply rewritten as a linear term:

$$\sum_{d=1}^{D} h^d \cdot y_i^d \qquad \text{with } h^d = f(d+1)$$

The Discretizing by Degree model (DDF) is presented in Fig. 2. Constraints (11) are consistency constraints for the new variables and state that if $y_i^d = 1$ for a given i and d then $y_i^p = 0$, $\forall p \neq d$. Notice that for nodes with degree equal to 1 we have $y_i^d = 0$, $\forall\, d = 1, \ldots, D$. Constraints (10) are the new coupling constraints and together with constraints (11) guarantee that, if the degree of a node i is greater than 1, the index d of the discretized variable with value equal to 1 associated with that node must be equal to the degree of the node minus 1.

$$(DDF)\ min \quad \sum_{e \in E} c_e\, x_e + \sum_{i \in V} \sum_{d=1}^{D} h^d \cdot y_i^d \tag{9}$$

$$s.to: \quad \{x_e, e \in E\} \in ST \tag{3}$$

$$\sum_{e \in E(i)} x_e = 1 + \sum_{d=1}^{D} d \cdot y_i^d \qquad \forall\, i \in V \tag{10}$$

$$\sum_{d=1}^{D} y_i^d \leq 1 \qquad \forall\, i \in V \tag{11}$$

$$x_e \in \{0,1\} \qquad \forall\, e \in E \tag{6}$$

$$y_i^d \in \{0,1\} \qquad \forall\, i \in V,\, d = 1, \ldots, D \tag{12}$$

Figure 2: Linear Formulation discretizing by degree

Next, following [8], we show how to construct rather intuitive inequalities using the new variables. By summing (10) over all nodes and by using the well-known fact that the sum of the node degrees in a graph is equal to twice the number of edges ($2 \cdot (n-1)$ in a spanning tree), we obtain the redundant constraint:

$$\sum_{d=1}^{D} d \cdot \sum_{i \in V} y_i^d = n - 2 \tag{13}$$

Dividing (13) by $g = 2, \ldots, D$ and by rounding down (rounding up) each coefficient in the left-hand side term and subsequently by rounding down (rounding up) the right-hand side term we obtain the following two sets of valid inequalities (note that $\left\lfloor \frac{d}{g} \right\rfloor = 0$, $\forall\, d < g$):

$$\sum_{d=g}^{D} \left\lfloor \frac{d}{g} \right\rfloor \cdot \sum_{i \in V} y_i^d \leq \left\lfloor \frac{n-2}{g} \right\rfloor \qquad g = 2, \ldots, D \tag{14}$$

$$\sum_{d=1}^{D} \left\lceil \frac{d}{g} \right\rceil \cdot \sum_{i \in V} y_i^d \geq \left\lceil \frac{n-2}{g} \right\rceil \qquad g = 2, \ldots, D \tag{15}$$

A careful examination of (14) and (15) shows that some of these inequalities are dominated by others in the same family since we obtain many such inequalities with the same right-hand side. Thus we only need to consider the inequality with the "stronger" left-hand side. As our results will show the previous inequalities tighten the linear programming relaxation of the original model. We shall denote by $(DDF+)$ the discretized model augmented with these constraints.

Finally, to motivate a model with a (hopefully) improved linear programming relaxation we first add the redundant constraint (13) to the discretized model. By subsequently removing the linking constraints (10), we obtain two subproblems, one in the x variables, a spanning tree problem, and the other in the y variables, a knapsack-like problem with an equality constraint (in fact, a minimization version of a *Multiple-choice Knapsack problem*, see [9]). We can view the whole problem as an intersection of two integer subproblems. This decomposition suggests at once a new model, where we will replace the knapsack constraints (11), (12) and (13), by a linear system describing the associated convex hull. It is well known that it is hard to write down such a system (this follows from the fact that the Knapsack problem is *NP-hard*). However we can use a pseudo-polynomial extended formulation that is obtained by writing the shortest path equations of the underlying dynamic program for solving the knapsack problem and adequately combining it with the discretized formulation. In the graph associated to this dynamic program, each node is characterized by two parameters $[i, s]$ such that:

- i, which denotes a given original node ($i = 0$ (ficticious node), $1, \ldots, n$)
- s, the sum of the value $d(i) - 1$ for all original nodes $1, \ldots, i$ ($s = 0, \ldots, n-2$)

With the parameter s defined as above any node $[i, s]$ with $s > i \cdot D$ or $s < (n-2) - D \cdot (n-i)$ are forbidden in this graph representation. Each arc of this graph begins in a node $[i-1, s]$ and ends in a node $[i, t]$ and represents the decision of setting the degree of the original node i to $t - s + 1$. Any *shortest path* from node $[0, 0]$ to node $[n, n-2]$ represents a feasible assignment of node degrees. Consider the *shortest path variables* z_i^{st} indicating whether arc ([i,s],[i+1,t]) is in the shortest path from node $[0, 0]$ to node $[n, n-2]$ ($\forall\ i = 0, \ldots, n-1$; $s, t = 0, \ldots, n-2$ and $t - s \leq D$). In order to obtain a formulation for our problem we need to relate the shortest path variables $z_i^{s,s+d}$ with the y_i^d variables. Note that, if for a given node i, we have $y_i^d = 1$ for some d then there exists only one $s \in \{0, \ldots, n-2-d\}$ such that $z_{i-1}^{s,s+d} = 1$ and all other variables for node i are zero. If $y_i^d = 0, \forall\ d = 1, \ldots, D$ then there exists only one $s \in \{0, \ldots, n-2-d\}$ such that $z_{i-1}^{s,s} = 1$ and all other variables for node i are null. This implies that the relation between both sets of variables is as follows:

$$y_i^d = \sum_{\substack{s=0 \\ (s+d) \leq D}}^{n-2} z_{i-1}^{s,s+d} \quad , \forall i = 1, \ldots, n,\ d = 1, \ldots, D \tag{16}$$

The knapsack shortest-path reformulation is depicted in Fig. 3 where $Path_{[0,0] \mapsto [n,n-2]}$ represents the set of all paths from node $[0, 0]$ to node $[n, n-2]$ in the graph associated to the knapsack dynamic program.

As the shortest path system is integral, the linear programming relaxation of the new model gives a lower bound that is at least as good as the linear programming relaxation bound of the previously mentioned improved model since the shortest path system does not provide a complete linear description of the knapsack convex hull, that is

$$V(\overline{KSSPF}) \geq V(\overline{DDF+})$$

An extended version of this paper is being developed by the authors including other models similar to the multicommodity network models with piecewise linear costs [1].

$$(KSSPF)\ min \sum_{e \in E} c_e\, x_e + \sum_{i=1}^{n} \sum_{d=1}^{D} h^d \cdot \sum_{\substack{s=0 \\ (s+d) \leq n-2}}^{n-2} z_{i-1}^{s,s+d} \tag{17}$$

$$s.to: \{x_e, e \in E\} \in ST \tag{3}$$

$$\{z_i^{s,t}\} \in Path_{[0,0] \mapsto [n,n-2]} \tag{18}$$

$$\sum_{e \in E(i)} x_e = 1 + \sum_{d=1}^{D} d \sum_{\substack{s=0 \\ s+d \leq n-2}}^{n-2} z_{i-1}^{s,s+d} \qquad \forall\, i \in V \tag{19}$$

$$x_e \in \{0,1\} \qquad \forall\, e \in A \tag{6}$$

Figure 3: Knapsack Shortest Path Reformulation

3. Preliminary results

The results presented in this section are only preliminary results refering to the lower bounds obtained by the linear relaxation programming of the models presented in the previous section. A hybrid GRASP/VND metaheuristic was also developed for this problem and is available in Duhamel and Souza [6]. To test both formulations (DDF, $DDF+$ and $KSSPF$) we used a set of randomly generated instances. The number of nodes has been set to 25 and 50 nodes and the graph density has been set to 25%, 50% and 100%. For the maximum node degree we used three different values $D + 1 = 4, 5, 6$. The edge costs have been randomly generated in the interval [1;100]. The module capacity has been set to $B = 3$ and cost parameters P_1 and P_2 have been set to 20 and 10 respectively leading to a degree node cost fuction: $f(1) = 0,\ f(2) = f(3) = 30,\ f(4) = f(5) = f(6) = 40$. Both formulations were coded in C using CPLEX technology in a Pentium IV processor 3.2 GHz 1.00 Gb RAM.

The results are shown in Table 1 where the first three columns represent the number of nodes, number of edges and maximum node degree. The forth column is the optimal integer value. Next, each column triplet represents the optimal value of the linear programming relaxation bound, the gap to the optimal integer value and the CPU time in seconds. For any instance (n, m), the gap increased with the maximum degree. Fixing the maximum degree, the gap decreased as the density of the graph increased. Comparing the three linear programming bounds, the added valid inequalities present in $(DDF+)$ reduced the gap for all instances with $n = 25$ but did not work as well for $n = 50$; the gap was reduced only in those instances with $D + 1 = 6$. Formulation $(KSSPF)$ reduced the gap even further only in instances where $D + 1 = 6$.

References

[1] Belotti, B., L. Brunetta and F. Malucelli, "Multicommodity network design with discrete node costs," to appear in *Networks*.

[2] Correia, I., L. Gouveia and F. Saldanha da Gama, "Solving the variable size bin packing problem with discretized formulations," to appear in *Computers and Operations Research*.

[3] Croxton, K., B. Gendron and T. Magnanti, "A comparison of Mixed-Integer Programming Models for Non-Convex Piecewise Linear Cost Minimization Problems," *Management Science*, volume 49, pages 1268-1273, 2003.

[4] Croxton, K., B. Gendron and T. Magnanti, "Variable Disaggregation in Network Flow Problems with Piecewise Linear Costs," to appear in *Operations Research*.

[5] Cunha, A. and A. Lucena, "Algorithms for the degree-constrained minimum spanning tree problem," *Eletronic Notes in Discrete Mathematics*, volume 19, pages 403-409, 2005.

				Linear Lower bounds								
n	m	$D+1$	Int	DDF	gap	CPU	$DDF+$	gap	CPU	$KSSPF$	gap	CPU
25	75	4	1665	1592	4.4	0.22	1595	4.2	0.14	1595	4.2	0.38
		5	1623	1498	7.8	0.20	1508	7.1	0.14	1508	7.1	0.38
		6	1589	1436	9.7	0.22	1447	8.9	0.14	1452	8.6	0.36
	150	4	1372	1331	3.0	0.31	1334	2.8	0.22	1334	2.8	0.67
		5	1311	1237	5.6	0.30	1247	4.9	0.19	1247	4.9	0.58
		6	1282	1176	8.3	0.28	1187	7.4	0.19	1192	7.0	0.48
	300	4	1476	1446	2.0	0.89	1449	1.8	0.78	1449	1.8	4.20
		5	1429	1358	5.0	0.94	1368	4.3	0.92	1368	4.3	3.49
		6	1403	1302	7.2	1.27	1313	6.4	0.94	1318	6.0	1.51
50	300	4	3336	3242	2.8	11.86	3242	2.8	11.31	3242	2.8	51.74
		5	3191	3017	5.5	10.25	3017	5.5	11.70	3017	5.5	21.83
		6	3109	2884	7.2	13.26	2895	6.9	9.58	2900	6.7	67.06
	600	4	3063	2990	2.4	29.91	2990	2.4	28.88	2990	2.4	76.63
		5	2948	2806	4.8	26.58	2806	4.8	19.06	2806	4.8	75.49
		6	2900	2697	7.0	23.80	2708	6.6	39.94	2713	6.4	71.11
	1225	4	2498	2459	1.6	300.31	2459	1.6	274.33	2459	1.6	458.66
		5	2380	2286	3.9	204.04	2286	3.9	127.95	2286	3.9	281.66
		6	2332	2181	6.5	168.19	2192	6.0	161.52	2197	5.8	344.55

Table 1: Results for 25 and 50 nodes instances

[6] Duhamel, C. and M. Souza, “A GRASP heuristic for the Node Degree Cosntrained Minimum Spanning Tree Problem with Node Degree Costs,” submitted to the *Proceedings from the International Network Optimization Conference*, 2007.

[7] Gouveia, L., “A 2n constraint formulation for the capacitated minimal spaning tree problem,” *Operations Research*, volume 43, pp. 130-141, 1995.

[8] Gouveia, L. and F. Saldanha da Gama, “On the capacitated concetrator location problem: a reformulation by discretization,” *Computers and Operations Research*, volume 33, pp. 1242-1258, 2006.

[9] Kellerer, H., U. Pferschy and D. Pisinger, *Knapsack Problems*, Springer , 2004.

[10] Kruskal, J. “On the shortest spanning subtree of a graph and the travelling salesman problem,” In *Proceedings of the Amercian Mathematics Society*, volume 7-1, pages 48-50, 1956.

[11] Magnanti, T., P. Mirchandani and R. Vachani, “Modeling and solving the two-facility capacitated network loading problem,” *Operations Research*, volume 43, pages 142-157, 1995.

[12] Magnanti, T. and L. Wolsey, *Handbooks in Operational Research and Management Science: Optimal Trees*, M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser, 1995.

[13] Prim, R., “Shortest connection networks and some generalizations,” In *Bell System Technical Journal*, volume 36, pages 1389-1401, 1957.

[14] Uchoa, E., R. Fukasawa, J. Lysgaard, A. Pessoa, M. Poggi de Aragão and D. Andrade, “Robust Branch-Cut-and-Price for the Capacitated Minimum Spanning Tree Problem over a Large Extended Formulation,” to appear in *Mathematical Programming*