# Spanning Trees with Node Degree Dependent Costs and Knapsack Reformulations

Luís Gouveia* Pedro Moura*

* *CIO-DEIO, Bloco C6-Piso 4, Faculdade de Ciências da Universidade de Lisboa*
*Cidade Universitária, Campo Grande, 1749-016 Lisboa, Portugal.*
{legouveia,pmmoura}@fc.ul.pt.
*Sponsored by FCT, research grants POCTI-ISFL-1-152 and POSI/CPS/41459/2001*

---

**Abstract**

The Degree constrained Minimum Spanning Tree Problem (DMSTP) consists in finding a minimal cost spanning tree satisfying the condition that every node has degree no greater than a fixed value. We consider a generalization with a more general objective function including modular costs associated to the degree of each node. We present linear programming models together with some valid inequalities and compare their respective linear programming relaxations using instances with up to 100 nodes.

**Keywords**: *Spanning Trees, Discretized Reformulation, Knapsack Reformulation.*

---

## 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 solved in polynomial time (see, for instance, [12, 16]). 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 Problem - DMSTP - (see the recent work of Cunha and Lucena, [6], and the references inside) which is known to be *NP-hard*. This constraint is usually motivated by the need to impose a limit on the number of ports in each node. However, the studies on the DMSTP ignore the intuitive and realistic generalization having an objective function that also considers costs on the nodes which depend on the value of the corresponding degree. It is quite intuitive to assume that nodes with degree 1 should be cheaper than nodes of degree greater than 1 and that the higher the degree is the greater is the corresponding node cost. In this presentation we describe one such generalization where the node costs are motivated in the context of a telecommunications network.

The node degree dependent costs are associated to multiplexing equipment that needs to be installed at the nodes of the network that are connected to more than one link. This equipment consists of a *switch matrix* and *interface modules*. Whenever a packet reaches one node in these conditions, the switch matrix needs to make a decision (based on the information on the packet's header) through which port (of the interface modules installed on that node) the packet should be sent through. Then the packet is placed in the respective queue. Each interface module has a fixed number of ports that may be, or not, all in use. More than one interface module can be installed in every node permitting the increase of the number of links connected to that node. The degree dependent cost of any non-leaf node is then given by the *switch matrix cost* plus a *cost per interface module* (dependent on the number of interface modules installed) and is defined as, $f^k = P_1 + k \cdot P_2$, $\forall k \geq 1$. The parameters $P_1$ and $P_2$ are respectively, the *switch matrix cost* and *interface module cost*, $k$ is the number of modules needed to support all the rerouting connections at the node. Let $B$ denote the module capacity (the fixed number of ports in any interface module), $deg(i)$ denote the degree of node $i \in V$ in a given feasible solution and let $D+1$ denote the maximum number of links that a node can support (that is, $deg(i) \leq D+1$ for any node $i$). Then the maximum number of modules that need to be installed in any node $i$ is $K = \lceil \frac{D+1}{B} \rceil$. We define an objective function that involves the term on the sum of the edge costs as well as a term on the node degrees, $\sum_{i \in V} f(deg(i))$, such that:

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

The cost function $f(deg(i))$ is typical of the so-called *Network Loading Problems* (see for instance [4, 5, 7, 13]). In these problems, the *loading* constraints are usually imposed on the arcs of the network. Problems including discrete node costs as well, are described and studied by Höller *et al.* [10] and by Höller and Voß [11]. In our work we focus only on discrete node costs, similarly to the study described in Belotti *et al.* [1]. However, our study focus on tree networks.

# 2 Integer Programming Formulations

Although the original problem has been defined with symmetric edge costs, we will focus our presentation on directed formulations since it is known that directed formulations lead, in general, to models with a tighter LP bound (see, for instance [14]). To view the problem as a directed one, we consider the problem of finding a minimum cost arborescence rooted away, for instance, from node 1, in a directed graph $G = (V, A)$ where $A$ is the set of arcs such that for each edge $e = \{i, j\} \in E$, we have two arcs $(i, j)$, $(j, i) \in A$ with the same cost as the original edge. Edges $e = \{1, j\} \in E$ are replaced only by a single arc $(1, j) \in A$. We also let $A^+(i)$ ($A^-(i)$) denote the set of all arcs diverging from (converging to) node $i$ and $\Phi(S) = \sum_{(i,j) \in S} \phi_{ij}$. With respect to a generic model $(P)$, let $(\overline{P})$ and $V(P)$ be, respectively, the LP relaxation and the optimal value of model $(P)$.

## 2.1 Integer Non-Linear Programming Formulation

Let $\langle DST \rangle$ denote the convex hull of incidence vectors of directed spanning trees rooted at node 1. Consider binary variables $x_{ij}$ indicating whether arc $(i, j)$ is in the solution and integer variables $u_i$ such that $u_i = deg(i) - 1$ in the solution. The problem can be modeled as an integer non-linear program $(NLF)$ (see Figure 1).

Constraints (2) are given in a generic form and can be represented by several equivalent sets of linear constraints (see for instance [14]) that either involve only the $x_{ij}$ variables (leading, in general, to exponential sized sets of inequalities) or involve other variables as well (leading, in general, to compact formulations). One well known such formulation is the so-called multicommodity flow formulation and will be used in our computations. Constraints (3a) and (3b) define the outdegree of every node and relate the two sets of variables. Since the solution is viewed as directed away from node 1 and one single arc is incident into every node, except the root node, the given constraints differentiate the two cases (root node and non-root nodes). Constraints (4) are upper bound degree constraints and constraints (5) define the domain of the $u_i$ variables.

$$
\begin{aligned}
&(NLF)\ min \quad \sum_{(i,j)\in A} c_{ij}\, x_{ij} + \sum_{i\in V} f(u_i+1) && (1)\\
&s.to: \quad x \in \langle DST \rangle && (2)\\
&X(A^+(i)) = u_i \qquad \forall\, i \in V\backslash\{1\} && (3a)\\
&X(A^+(1)) = 1 + u_1 && (3b)\\
&u_i \leq D \qquad \forall\, i \in V && (4)\\
&u_i \ \in \mathbb{N}_0 \qquad \forall\, i \in V && (5)
\end{aligned}
$$

Figure 1: Non-linear Formulation

## 2.2 Integer Linear Programming Formulations

### 2.2.1 Discretized Formulations Part I: Discretizing the Node Variables

This formulation 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. We obtain the discretized model from the previous non-linear model, by using

$$u_i = \sum_{d=1}^{D} d \cdot y_i^d \quad ,\forall\, i \in V \tag{6}$$

to replace each occurrence of the "old" variables $u_i$ by the right-hand side of the previous relation and by including the consistency constraints (9). After the transformation, the non-linear term $f\left(\sum_{d=1}^{D} d \cdot y_i^d + 1\right)$ in the objective function can be rewritten as the linear term on the objective function (7) (with $h^d = f(d+1)$). The new model $(DNV)$ is shown in Figure 2.

$$
\begin{aligned}
&(DNV)\ min \quad \sum_{(i,j)\in A} c_{ij}\, x_{ij} + \sum_{i\in V}\sum_{d=1}^{D} h^d \cdot y_i^d && (7)\\
&s.to: \quad x \in \langle DST \rangle && (2)\\
&X(A^+(i)) = \sum_{d=1}^{D} d \cdot y_i^d \qquad \forall\, i \in V\backslash\{1\} && (8a)\\
&X(A^+(1)) = 1 + \sum_{d=1}^{D} d \cdot y_1^d && (8b)\\
&\sum_{d=1}^{D} y_i^d \leq 1 \qquad \forall\, i \in V && (9)\\
&y_i^d \in \{0,1\} \qquad \forall\, i \in V, d = 1,\ldots,D && (10)
\end{aligned}
$$

Figure 2: Linear Formulation - Discretizing the Node Variables

Constraints (8a) and (8b) are the new coupling constraints and guarantee that for nodes with degree equal to 1 we have $y_i^d = 0, \forall\, d = 1, \ldots, D$. Together with the consistency constraints (9) these constraints guarantee that if the degree of a node $i$ is equal to $d^* + 1 > 1$, then $y_i^{d^*} = 1$ and $y_i^p = 0, \forall\, p \neq d^*$.

This idea of substituting an integer variable with a set of binary variables with an index indicating the possible values of the original integer value has already been applied to other problems (see, for instance, [2, 3, 8, 9, 17]). In these works, the main reason for performing this change of variable set is to create new and effective sets of valid inequalities. However, as noted in [8] and then used in [9], the new variables permit us to model versions of the original problem with non linear costs. The problem studied here exemplifies this situation since the discretized variables permits us to model the more general cost function. Note also that due to the degree upper limit, the $(DNV)$ model has fewer discretized variables than the discretized models in the other previously referred works (which was a drawback of the approach in these cases).

The linear programming relaxation of the $(DNV)$ model is, in general, quite weak. However, the structure of constraints (8a) suggests the following set of quite intuitive valid inequalities that significantly improve the linear programming relaxation of $(DNV)$:

$$x_{ij} \leq \sum_{d=1}^{D} y_i^d \qquad \forall\ (i,j) \in A,\ i \neq 1 \tag{11}$$

Note that the intuition for adding the new constraints to $(DNV)$ is similar to adding so-called strong location cuts in capacitated location models. We will denote by $(DNV+)$ the $(DNV)$ model with the new set of constraints.

#### 2.2.2 Spanning Tree/Knapsack reformulation

Consider again the non-linear formulation $(NLF)$. Summing up constraints (3a) and (3b) and by using $\sum_{(i,j)\in A} x_{ij} = n - 1$ (which is implied by $\langle DST \rangle$) we obtain:

$$\sum_{i \in V} u_i = n - 2 \tag{12}$$

Adding this redundant constraint to model $(NLF)$ permits us to view the problem as being composed of two subproblems: a directed spanning tree problem in the $x$ variables, and a bounded knapsack problem with an equality constraint (see [15]) in the $u$ variables, with an additional set of constraints, (3a) and (3b), linking the variables of the two subproblems. We can replace the constraints for the knapsack subproblem by the path equations in an adequate graph. Let $\hat{G} = (\hat{V}, \hat{A})$ be the *extended graph* associated with this path representation. Each node $i_s \in \hat{V}$ is characterized by two integer parameters $i, s$ such that $i$ denotes a given original node ($i = 0$ (ficticious node), $1 \ldots, n$) and $s$ denotes the sum $\sum_{k=1}^{i} u_k$ ($s = 0, \ldots, n-2$). Each arc in $\hat{A}$ is of the form $\left(i-1_s, i_t\right)$ (w.l.o.g. will be represented as $(i_{s,t})$) and represents the decision of setting the degree of node $i$ in the original graph to $t - s + 1$. The arc set is then explicitly defined as $\hat{A} = \{\, (i_{s,t}) : i-1_s\,,\, i_t \in \hat{V}, 0 \leq t - s \leq D \,\}$. Any *path* from node $0_0$ to node $n_{n-2}$ in the extended graph represents a feasible assignment of degrees to all nodes in the original graph. The *path variables* $w_i^{st}$ indicate whether arc $(i_{s,t}) \in \hat{A}$ is in the shortest path from node $0_0$ to node $n_{n-2}$. Figure 3 shows the equations that model a path in the extended graph representation.

The path variables $w_i^{st}$ and the degree variables $u_i$ are related as follows:

$$u_i = \sum_{(i_{s,t}) \in \hat{A}} (t - s) \cdot w_i^{st} \quad , \forall i \in V \tag{17}$$

$$W(\hat{A}^+(0_0)) = 1 \tag{13}$$
$$W(\hat{A}^-(i_s)) = W(\hat{A}^+(i_s)) \qquad \forall\ i_s \in \hat{V}; \quad i = 1, \ldots, n-1 \tag{14}$$
$$W(\hat{A}^-(n_{n-1}) = 1 \tag{15}$$
$$w_i^{st} \in \{0, 1\} \qquad \forall\ (i_{s,t}) \in \hat{A} \tag{16}$$

Figure 3: Linear system associated with the Path representation.

Let $\langle K/P0 \rangle = \{(u, w)$ satisfying $(13), (14), (15), (16), (17)$ and $(5)\}$ be the set of feasible solutions of the knapsack subproblem stated in terms of paths of the *extended graph*. The interest of this reformulation is that the path variables give information on the value of the degree of node $i$ (more precisely, if $w_i^{st} = 1$ then, the degree of node $i$ is equal to $t - s + 1$) and we can rewrite the objective function of the model in a linear way leading to the the Spanning Tree/Knapsack reformulation depicted in Figure 4.

$$(ST/K0)\ min \qquad \sum_{(i,j)\in A} c_{ij}\, x_{ij} + \sum_{i=1}^{n} \sum_{(i_{s,t})\in \hat{A}} h^{t-s} \cdot w_i^{st} \tag{18}$$

$$s.to: \quad x \in \langle DST \rangle \tag{2}$$
$$(u, w) \in \langle K/P0 \rangle \tag{19}$$
$$\sum_{(i,j)\in A^+(i)} x_{ij} = u_i \qquad \forall\ i \in V \backslash \{1\} \tag{3a}$$
$$\sum_{(1,j)\in A^+(1)} x_{1j} = 1 + u_1 \tag{3b}$$

Figure 4: Spanning Tree/Knapsack Reformulation of model $(NLF)$

In our presentations we prove that the linear programming bounds of the knapsack model $(ST/K0)$ are never worse than the linear programming bounds of model $(DNV)$.

**Result 2.1** $V(\overline{ST/K0}) \geq V(\overline{DNV})$

We can also tighten the linear programming bound of the $(ST/K0)$ model by adding the following constraints:

$$x_{ij} \leq \sum_{d=1}^{D} \sum_{(i_{s,s+d})\in \hat{A}} w_i^{s,s+d} \qquad \forall\ (i,j) \in A,\ i \neq 1 \tag{20}$$

We denote by $(ST/K0+)$ the $(ST/K0)$ model with the new set of valid inequalities. We show that in a certain sense, these new inequalities are the same as the inequalities (11) presented at the end of section 2.2.1. Our results will show that the linear programming bounds of the knapsack model $(ST/K0)$ are never worse (better in several cases) than the linear programming bounds produced by the $(DNV)$ model. A similar dominance relationship arises between the models $(ST/K0+)$ and $(DNV+)$.

**Result 2.2** $V(\overline{ST/K0+}) \geq V(\overline{DNV+})$

# References


[1] P. Belotti, L. Brunetta and F. Malucelli, *Multicommodity network design with discrete node costs*, Networks, **49** (2007) pp. 90–99.

[2] M. Constantino and L. Gouveia, *Reformulation by discretization: application to Economic Lot Sizing*, OR letters **35(5)** (2007) pp. 645–650.

[3] I. Correia, L. Gouveia and F. Saldanha da Gama, *Solving the variable size bin packing problem with discretized formulations*, Computeres and Operations Research **35(6)** (2008) pp. 2103–2113.

[4] K. Croxton, B. Gendron and T. Magnanti. *A comparison of Mixed-Integer Programming Models for Non-Convex Piecewise Linear Cost Minimization Problems*, Management Science **49** (2003) pp 1268–1273.

[5] K. Croxton, B. Gendron and T. Magnanti. *Variable Disaggregation in Network Flow Problems with Piecewise Linear Costs*, Operations Research **55** (2007) pp 146–157.

[6] A. Cunha and A. Lucena. *Lower and upper bounds for the Degree-constrained Minimum Spanning Tree Problem*, Networks **50(1)** (2007) pp 55–66.

[7] A. Frangioni and B. Gendron, *0-1 Reformulations of the Network Loading Problem*, Discrete Applied Mathematics (to appear).

[8] L. Gouveia, *A 2n constraint formulation for the capacitated minimal spanning tree problem*, Operations Research **43** (1995) pp. 130–141.

[9] L. Gouveia and F. Saldanha da Gama, *On the capacitated concentrator location problem: a reformulation by discretization*, Computers and Operations Research **33** (2006) pp. 1242–1258.

[10] H. Höller, B. Melián, S. Voß, it Applying the pilot method to improve VNS and GRASP metaheuristics for the design of SDH/WDM networks, European Journal of Operational Research **191(3)** (2008) pp. 691–704.

[11] H. Höller, S. Voß, *A heuristic approach for combined equipment-planning and routing in multi-layer SDH/WDM networks*, European Journal of Operational Research **171(3)** (2006) pp. 787–796.

[12] J. Kruskal, *On the shortest spanning subtree of a graph and the travelling salesman problem*, Proceedings of the Amercian Mathematics Society **7(1)** (1956) pp. 48–506.

[13] T. Magnanti, P. Mirchandani and R. Vachani, *Modeling and solving the two-facility capacitated network loading problem*, Operations Research **43** (1995) pp. 142–157.

[14] T. L. Magnanti 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.

[15] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, 1990.

[16] R. Prim, *Shortest connection networks and some generalizations*, Bell System Technical Journal **36** (1957) pp. 1389–1401.

[17] E. Uchoa, 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*, Mathematical Programming **112(2)** (2008) pp. 443–472.