MODELLING HOP-CONSTRAINED AND DIAMETER-CONSTRAINED MINIMUM SPANNING TREE PROBLEMS AS STEINER TREE PROBLEMS OVER LAYERED GRAPHS

Luis Gouveia
DEIO/CIO – Faculdade de Ciências da Universidade de Lisboa
Bloco C/2 - Campo Grande, Cidade Universitária, 1749-016, Lisboa
legouveia@fc.ul.pt

Luidi Simonetti
PESC/COPPE – Universidade Federal do Rio de Janeiro
Cidade Universitária, CT, Bloco H, Sala 319, 21941-972, Rio de Janeiro, RJ
luidi@cos.ufrj.br

Eduardo Uchoa
Departamento de Engenharia de Produção – Universidade Federal Fluminense
Rua Passo da Pátria, 156, São Domingos, 22210-240, Niterói, RJ
uchoa@producao.uff.br

Abstract

The Hop-Constrained Minimum Spanning Tree Problem (HMSTP) arises in the design of centralized telecommunication networks with quality of service constraints. We show that the HMSTP is equivalent to a Steiner Tree Problem (STP) in an adequate layered graph. We prove that the directed cut formulation for the STP defined in the layered graph, dominates the best previously known formulations for the HMSTP. We then show how cuts in the extended layered graph space can be projected into new families of cuts in the original design space. We also adapt the proposed approach for the Diameter-Constrained Minimum Spanning Tree Problem (DMSTP). Computational results with a branch-and-cut algorithm show that the proposed method is significantly better than previously known methods on both problems.

Keywords: Networks/Graphs: tree algorithms, Integer Programming: formulations, cutting planes

Resumo

O Problema da Árvore Geradora Mínima com Restrições de Salto (HMST) surge no projeto de redes de comunicação centralizadas com restrições de qualidade de serviço. Mostramos que o HMST é equivalente a um Problema da Árvore de Steiner (STP) sobre um grafo em camadas

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

adequado. Provamos que a formulação de cortes direcionados para o STP aplicada ao grafo de camadas domina as melhores formulações conhecidas para o HMST. Também mostramos que cortes definidos sobre o grafo em camadas podem ser projetadas em novas famílias de cortes no espaço de variáveis original. A abordagem também pode ser adaptada para o O Problema da Árvore Geradora Mínima com Restrições de Diâmetro (DMST). Os resultados computacionais obtidos em ambos os problemas são significativamente melhores do que os obtidos com outros métodos conhecidos.

Palavras-Chave: Redes/Grafos: algoritmos para árvores, Programação Inteira: formulações, planos de corte

1 Introduction

The Hop-constrained Minimum Spanning Tree Problem (HMSTP) is defined as follows: given a graph $G = (V, E)$ with node set $V = \{0, 1, \ldots, n\}$ and edge set E as well as a positive cost c_e associated with each edge e of E and a natural number H, we wish to find a spanning tree T of the graph with minimum total cost and such that the unique path from a specified root node, node 0, to any other node has no more than H hops (edges).

The HMSTP is NP-hard because it contains as a particular case (the case with $H = 2$) a NP-Hard version of the Simple Uncapacitated Facility Location problem (see [11, 4]). [28] have shown that the HMSTP is not in APX, i.e., the class of problems for which it is possible to have polynomial time heuristics with a constant-factor approximation bound. The HMSTP models the design of centralized telecommunication networks with quality of service constraints. The root node represents the site of a central processor and the remaining nodes represent terminals that are required to be linked to the central processor. The hop constraints guarantee a certain level of service with respect to some performance constraints such as availability and reliability (see [39]). Availability is the probability that all the transmission lines in the path from the root node to the terminal are working. Reliability is the probability that a session will not be interrupted by a link failure. These probabilities decrease with the number of links in the path, implying that paths with fewer hops have a better performance. Centralized terminal networks are also usually implemented with multidrop lines for connecting the terminals with the center. In such networks, node processing times dominate over link delays and fewer hops mean, in general, lower delays.

Lower bounding schemes for the HMSTP based on network-flow models have been suggested in [12, 14], and [19]. More recently, [6] propose a formulation involving only natural design variables and an exponential sized set of so-called "jump" constraints and propose a lower bound based on an appropriate Lagrangean relaxation. The recent survey by [8] summarizes these approaches. The reader is also referred to [22] for a survey of several network design problems with hop constraints and methods to solve them.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

The models described in the previous papers view the HMSTP problem as defined in the original graph (although some extended models also use variables corresponding to arcs in auxiliary layered graphs associated to hop-constrained shortest path problems, one for each non-root node). In this paper we propose a modelling approach for the HMSTP that views the whole problem as defined in a single layered graph. We will show that the HMSTP is equivalent to a Steiner tree problem (see, for instance, [26, 21]) in the layered graph. Thus, any known formulation for this problem can be used to solve the HMSTP. Our computational results are taken from an adaptation of the efficient branch-and-cut method proposed in [31], and which is based on the well known directed cut formulation for the Steiner tree problem. We will also show that the linear programming relaxation bounds of the directed cut formulation defined in the layered graph are at least as good (strictly better for several instances) than the linear programming bounds of best HMST formulations presented so far.

[8] have shown that for $H = 3$, the linear programming feasible set of the formulation used in [6] equals the projection of the linear programming feasible set of the extended formulation given in [14] into the design variable subspace. They have also shown that obtaining a similar equivalence result for $H \geq 4$ would prove to be much more complicated. Since our new formulation dominates Gouveia's formulation, even for $H = 3$, it is interesting to have some insight of what inequalities are implied by the linear programming relaxation of the new formulation that are not redundant in the linear programming relaxation of the older one. We provide a technique for combining inequalities from the new formulation in a such a way that they can be projected into new families of cuts in the original design space.

The methods provided in this paper can be easily adapted for the HMSTP variant where the hop constraint $H(i)$ depends on the node. This variant is useful on telecommunication network design applications where more important nodes need to be closer to the central node.

A related problem, the so-called Diameter-constrained Minimum Spanning Tree Problem (DMSTP) is defined as follows: given a graph $G = (V, E)$ with node set $V = \{1, \ldots, n\}$ and edge set E as well as a positive cost c_e associated with each edge e of E and a natural number D, we wish to find a spanning tree T of the graph with minimum total cost and such that the unique path from any node i to any node j has no more than D hops (edges). Note that in this variation, we constrain the path between each pair of nodes while in the previous one, only the paths from the special node are constrained. Several approaches for the DMSTP (see, for instance, [1, 16, 33, 20]) have used the properties of tree centers in order to transform the DMSTP into special versions of the HMSTP. In a similar way, here we adapt the new HMTSP method for the DMSTP. Although the adaptation given for the situations with even diameter is taken straightforwardly from the literature, we propose a new transformation for the odd diameter case that permits us to make efficient use of the layered graph approach.

Although the HMSTP and the DMSTP have symmetric edge costs, we will focus our presentation on so-called directed formulations. It is known that directed formulations lead, in general, to models with a tighter linear programming bound (see, for instance [27]). To view the HMSTP as a directed one, we consider the problem of finding a minimum cost arborescence rooted away from node 0. To define the directed models we consider the problem redefined in a directed graph $G = (V, A)$ and such that for each edge (i, j) in E, we have two arcs (i, j) and (j, i) in A with the same cost as the original edge. With respect to edges $(0, j)$ we consider only a single arc $(0, j)$. A similar construct will be done for the DMSTP, since in the proposed approaches, a dummy central node is created and we can see the tree as directed away from it. The proposed transformation can be easily adapted for the Diameter-constrained Minimum Steiner Tree Problem, where only a subset of the nodes, the required nodes, need to be connected by a tree respecting a diameter constraint.

The remainder of the paper is as follows. In Section 2 we describe the layered graph and the transformation of the HMSTP into a Steiner tree problem in the layered graph. In Section 3 we rewrite some of the well known formulations for the Steiner tree problem in the context of the layered graph. In Section 4 we make a brief survey of the formulations with the best linear programming bound presented in the literature and compare their linear programming bounds with the linear programming relaxation bounds of the new formulations. This section also presents new inequalities in the design space that are implied by the new formulations. In Section 5 we review the STP branch-and-cut method described in [31]. In Section 6 we describe an adaptation of the transformation and method for the related DMSTP, for both even or odd diameters. The computational experiments are given in Section 7.

2 Transforming the HMSTP into a Steiner Tree Problem in a Layered Graph

In this section we show how to transform the HMSTP into a Steiner tree problem in a layered graph. Consider a graph $G_L = (V_L, A_L)$ defined as follows:

$$\begin{aligned}
V_L &= \{0\} \cup \{(i,h) : 1 \leq h \leq H \text{ and } i \in V \setminus \{0\}\} \text{ and} \\
A_L &= A_0 \cup A_1 \cup A_2 \text{ where} \\
&\quad A_0 = \{((0),(j,1)) : (0,j) \in A\}, \\
&\quad A_1 = \{((i,h),(j,h+1)) : (i,j) \in A, i \neq 0, 1 \leq h \leq H-1\} \text{ and} \\
&\quad A_2 = \{((i,h),(i,H)) : i \in V \setminus \{0\}, 1 \leq h \leq H-1\}.
\end{aligned}$$

The required node set R is defined as the set $\{(i, H) : i \in V \setminus \{0\}\}$. The cost of the arcs in the sets A_0 and A_1 are equal to the costs of the corresponding arcs in the original graph and the cost of arcs in A_2 is equal to 0. Note that G_L is built by levels with the nodes of the original graph replicated H times. A node (i, h) in the layered graph is associated to node i being in depth h in the original graph (i.e., the path from node 0 to node i contains h arcs). Figure 1 illustrates on its right-hand side the layered graph corresponding to the graph shown on the left-hand side, for an instance with $H = 3$. The root node is depicted as a triangle, nodes of R are depicted as squares, the arcs in A_2 are dashed.

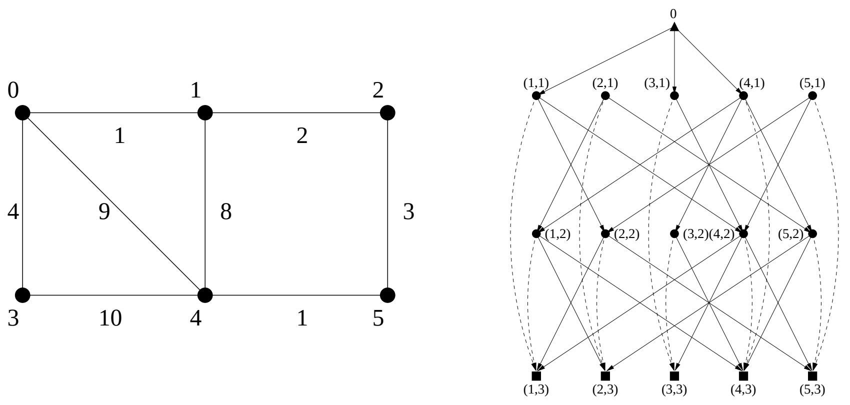

Figure 1: Layered Graph Transformation: original graph of an instance with $H = 3$ on the left-hand side and the corresponding layered graph on the right-hand side.

Consider the arcs in a minimum cost Steiner tree in G_L. Removing the arcs in A_2 and ignoring the indices h of the remaining arcs, one obtains a spanning tree with depth less or equal than H in the original graph and with the same cost. Conversely, every hop constrained spanning tree in the original graph corresponds to a Steiner tree with the same cost in the layered graph. One example of such pair of solutions is given in Figure 2.

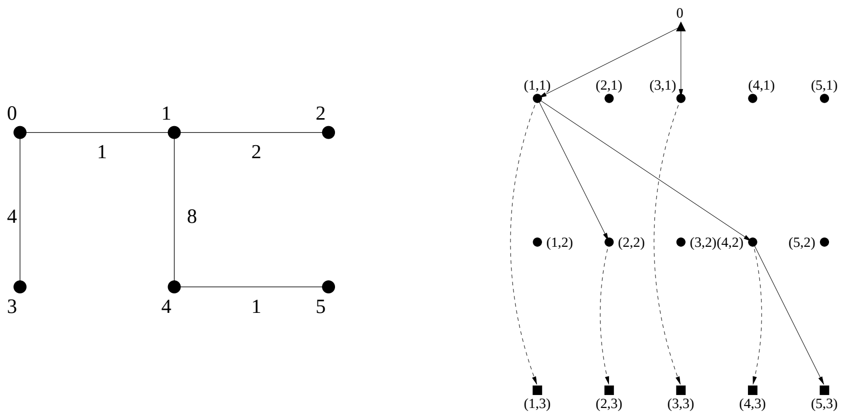

Figure 2: Optimal HMST in the original graph (cost 16) and its corresponding Steiner tree in G_L ($H = 3$)

3 The Formulations

The advantage of using the transformation described in the previous section is that by associating binary variables to each arc in the layered graph, we can use any model for the Steiner tree problem to provide a valid model for the HMSTP. More precisely, we define the following binary variables: (i) X_{0j}^1 for each arc $(0,(j,1))$ in A_0, (ii) X_{ij}^h for each arc $((i,h-1),(j,h))$ in A_1, and (iii) X_{jj}^h to each arc $((j,h-1),(j,H))$ in A_2. In the next two subsections we present three such models. The first one is an adaptation for the layered graph of the well known directed cut formulation given in [26] (see also [2]) for the Steiner tree problem. This model will be used in our computations. The second one is an equivalent (in the sense that the two models provide the same linear programming bound) network flow model (it can also be seen as [38] directed flow model adapted for the layered graph) and the third one is a simplified version of the previous one. The simplified network flow model will help us to relate the new models with other models presented in the literature.

3.1 The Directed Cut Model

Before introducing the cut model, we introduce some notation for describing some inequalities on the layered graph. Let $[V_L \setminus S, S]$ denote a cut in the graph such that $0 \notin S$ and $S \cap R \neq \{\emptyset\}$ and let $X[V_L \setminus S, S]$ denote the sum of the X_{ij}^h variables associated to arcs on that cut. The following model is nothing else than the directed cut Steiner model rewritten for the layered graph:

Model Hop-Cut

$$\min \sum_{j:(0,j)\in A} c_{0j} X_{0j}^1 + \sum_{(i,j)\in A, i\neq 0} c_{ij} \sum_{h=2}^{H} X_{ij}^h \tag{1}$$

$$\text{s.t.} \quad \sum_{i:(i,j)\in A} X_{ij}^H + \sum_{h=2}^{H} X_{jj}^h = 1, \qquad j \in V \setminus \{0\} \tag{2}$$

$$X[V_L \setminus S, S] \geq 1, \qquad S \cap R \neq \{\emptyset\}; 0 \notin S \tag{3}$$

$$X_{0j}^1 \in \{0,1\}, \qquad (0,j) \in A \tag{4}$$

$$X_{ij}^h \in \{0,1\}, \qquad (i,j) \in A;\ i \neq 0;\ h = 2,\ldots,H \tag{5}$$

$$X_{jj}^h \in \{0,1\}, \qquad j \in V \setminus \{0\};\ h = 2,\ldots,H \tag{6}$$

Constraints (2) guarantee that each required node is visited once. Constraints (3) are directed cut constraints and state that if some required nodes are in S, then any feasible solution for the problem must contain one arc, at least, in the cut $[V_L \setminus S, S]$. Note that when $|S \cap R| = 1$, constraints (3) are dominated by (2). However, when the arc costs are positive (as is the case of the problem studied here) we can omit constraints (2) as any optimal solution would satisfy these constraints as equalities. The formulation given above contains an exponential sized set of constraints (constraints (3)). However, they can be efficiently separated in polynomial time (see Section 5).

We note that formulations for the HMSTP using similar variables have been proposed before. In fact, the formulation presented in [13] for the HMSTP can be seen as specialized case of the [23] formulation for the Steiner tree problem defined in the layered graph. Gouveia's formulation uses so-called hop-ordering inequalities that are a hop-indexed version of a set of "connectivity" constraints proposed in the formulation by [23]. More recently, [3] have used use a similar formulation augmented with valid inequalities for a variation of the Steiner tree problem with Profits. An example from a different problem is taken from the hop-indexed formulation given in [18] for the capacitated minimum spanning tree. This formulation can be seen as a straightforward capacitated single commodity flow formulation defined in the layered graph. This layered graph construct has also been used for a weighted version of the HMSTP in [25]. This problem differs from the HMSTP in the sense that "distance" weights are attached to the arcs and the constraint requires that the sum of the arc weights on all paths from the root node do not exceed a given amount H. Finally, this brief survey on layered graph based models would not be complete without mentioning a few similar approaches on routing problems. Among them, we refer the well known formulation for the time dependent travelling salesman problem by [30] and which may be considered the first work using such a layered approach and a more recent approach by [10] for the Capacitated Vehicle Routing Problem, where the emphasis is on finding inequalities projected on the space of flow based formulations.

3.2 The Directed Network Flow Model

In order to compare the proposed formulation with others proposed in the literature, we follow the literature (see, for instance [26, 27]) and rewrite constraints (3) in a compact way by using network flows. For each $k \in V \setminus \{0\}$, we consider flow binary variables y_{ij}^{hk} indicating whether arc (i, j) is in position h in the path to node k in the original graph (or alternatively, whether arc $((i, h-1), (j, h))$ of G_L is in the path to node k in the Steiner tree solution) and variables y_{kk}^{hk} indicating whether node k is in position $h-1$ (or alternatively, whether arc $((k, h-1), (k, H))$ of G_L is in position h in the path to node k in the Steiner tree solution). Consider the following set of constraints.

$$\sum_{j \in V \setminus \{0\}} y_{0j}^{1k} = 1, \qquad k \in V \setminus \{0\} \tag{7}$$

$$y_{0i}^{1k} - \sum_{j \in V \setminus \{0,i\}} y_{ij}^{2k} = 0, \qquad i, k \in V \setminus \{0\}; i \neq k \tag{8}$$

$$y_{0k}^{1k} - y_{kk}^{2k} = 0, \qquad k \in V \setminus \{0\} \tag{9}$$

$$\sum_{j \in V \setminus \{0,i,k\}} y_{ji}^{hk} - \sum_{j \in V \setminus \{0,i\}} y_{ij}^{(h+1)k} = 0, \qquad i, k \in V \setminus \{0\}; i \neq k; h = 2, \dots, H-1 \tag{10}$$

$$\sum_{j \in V \setminus \{0,k\}} y_{jk}^{hk} - y_{kk}^{(h+1)k} = 0, \qquad k \in V \setminus \{0\}; h = 2, \dots, H-1 \tag{11}$$

$$\sum_{j \in V \setminus \{0,k\}} y_{jk}^{Hk} + \sum_{h=2}^{H} y_{kk}^{hk} = 1, \qquad k \in V \setminus \{0\} \tag{12}$$

$$y_{0j}^{1k} \geq 0, \qquad (0,j) \in A; k \in V \setminus \{0\} \tag{13}$$

$$y_{ij}^{hk} \geq 0, \qquad (i,j) \in A; i \neq 0; k \in V \setminus \{0\}; i \neq k; h = 2, \ldots, H \tag{14}$$

$$y_{jj}^{hj} \geq 0, \qquad j \in V \setminus \{0\}; h = 2, \ldots, H. \tag{15}$$

If we replace in the model Hop-Cut (3), these constraints together with the linking constraints

$$y_{0j}^{1k} \leq X_{0j}^{1}, \quad (0,j) \in A; k \in V \setminus \{0\} \tag{16}$$

$$y_{ij}^{hk} \leq X_{ij}^{h}, \quad (i,j) \in A; i \neq 0; k \in V \setminus \{0\}; i \neq k;\ h = 2, \ldots, H \tag{17}$$

$$y_{jj}^{hj} \leq X_{jj}^{h}, \quad j \in V \setminus \{0\}; h = 2, \ldots, H \tag{18}$$

we obtain another valid formulation for the HMSTP, denoted by Hop-NF in the remainder of the paper. Note that this formulation is nothing else than the well known directed network flow formulation for the Steiner tree problem defined in the layered graph (see again, [26, 27]). A simple application of the max cut/min flow theorem shows that the linear programming bound of the Hop-NF formulation equals the linear programming bound of the Hop-Cut formulation.

3.3 The Revised Directed Network Flow Model

Before comparing the linear programming relaxation of the new formulations with the linear programming relaxation of other formulations presented in the literature, we simplify the previous formulation. First, we observe that constraints (9) and (11) permit us to eliminate the variables y_{jj}^{hj}. Thus, the equalities (9), (11) and (15) can be removed from the model, constraints (12) become

$$\sum_{h=2}^{H} \sum_{j \in V \setminus \{0,k\}} y_{jk}^{hk} + y_{0k}^{1k} = 1, \quad k \in V \setminus \{0\} \tag{19}$$

(20)

and constraints (18) become

$$y_{0j}^{1k} \leq X_{jj}^{2}, \qquad j \in V \setminus \{0\} \tag{21}$$

$$\sum_{i \in V \setminus \{0,j\}} y_{ij}^{(h-1)j} \leq X_{jj}^{h}, \quad j \in V \setminus \{0\}; h = 3, \ldots, H. \tag{22}$$

Second, we observe that in the linear programming relaxation of the Hop-NF formulation, constraints (16), (17) when $j = k$ and constraints (18) are satisfied as equalities (and thus, the same happens with constraints (21) and

(22)). Combining these equalities gives

$$X^1_{0j} = X^2_{jj}, \qquad j \in V \setminus \{0\} \tag{23}$$

$$\sum_{i \in V \setminus \{0,j\}} X^{h-1}_{ij} = X^h_{jj}, \quad j \in V \setminus \{0\}; h = 3, \ldots, H \tag{24}$$

permitting us to eliminate the variables X^h_{jj}. Then, constraints (2) become:

$$\sum_{h=2}^{H} \sum_{i \in V \setminus \{0,j\}} X^h_{ij} + X^1_{0j} = 1, \quad j \in V \setminus \{0\} \tag{25}$$

We present next the Revised Hop-NF formulation after performing these simplifications.

Model Revised Hop-NF

$$\min \quad \sum_{j:(0,j)\in A} c_{0j}X_{0j}^{1} + \sum_{(i,j)\in A, i\neq 0} c_{ij}\sum_{h=2}^{H} X_{ij}^{h} \tag{26}$$

$$\text{s.t.} \quad \sum_{h=2}^{H}\sum_{i\in V\setminus\{0,j\}} X_{ij}^{h} + X_{0j}^{1} = 1, \qquad j \in V\setminus\{0\} \tag{27}$$

$$\sum_{j\in V\setminus\{0\}} y_{0j}^{1k} = 1, \qquad k \in V\setminus\{0\} \tag{28}$$

$$y_{0i}^{1k} - \sum_{j\in V\setminus\{0,i\}} y_{ij}^{2k} = 0, \qquad i,k \in V\setminus\{0\}; i\neq k \tag{29}$$

$$\sum_{j\in V\setminus\{0,i,k\}} y_{ji}^{hk} - \sum_{j\in V\setminus\{0,i\}} y_{ij}^{(h+1)k} = 0, \qquad i,k \in V\setminus\{0\}; i\neq k; h=2,\ldots,H-1 \tag{30}$$

$$\sum_{h=2}^{H}\sum_{j\in V\setminus\{0,k\}} y_{jk}^{hk} + y_{0k}^{1k} = 1, \qquad k \in V\setminus\{0\} \tag{31}$$

$$y_{0j}^{1k} \leq X_{0j}^{1}, \qquad (0,j)\in A; k\in V\setminus\{0\} \tag{32}$$

$$y_{ij}^{hk} \leq X_{ij}^{h}, \qquad (i,j)\in A; i\neq 0; k\in V\setminus\{0\}; h=2,\ldots,H \tag{33}$$

$$y_{0j}^{1k} \geq 0, \qquad (0,j)\in A; k\in V\setminus\{0\} \tag{34}$$

$$y_{ij}^{hk} \geq 0, \qquad (i,j)\in A; i\neq 0; k\in V\setminus\{0\}; h=2,\ldots,H \tag{35}$$

$$X_{0j}^{1} \in \{0,1\}, \qquad (0,j)\in A \tag{36}$$

$$X_{ij}^{h} \in \{0,1\}, \qquad (i,j)\in A; i\neq 0; h=2,\ldots,H \tag{37}$$

The reasoning given for obtaining this model shows that the linear programming bound of Hop-NF equals the linear programming bound of the revised Hop-NF formulation.

4 Surveying known Formulations

In this chapter we survey two formulations from the literature. One formulation differs from the ones presented here in the sense that the whole problem is modelled in the original graph but the underlying hop-constrained path problems are modelled on a layered graph similar to the one previously discussed. This formulation is the previously known formulation for the HMSTP with the best linear programming bound and here we prove that the new formulations presented in this paper produce a linear programming bound that it is at least as good (strictly better for several instances). The second formulation surveyed in this section has a weaker linear programming. However, it is, as far as we know, the only formulation for the HMSTP that uses one variable for each arc of the graph (although it uses an exponential sized set of so-called jump inequalities which are, in a certain sense, the analogue of the cut inequalities for hop constrained network design problems).

4.1 A Formulation Modelling the Underlying Path Subproblem in a Layered Graph

[14] has presented a formulation for the HMSTP that is quite similar to the previous NF-Hop and Revised NF-Hop formulations (this formulation will be denoted by Hop-MCF in the remainder of the text). The main difference is that the Hop-MCF formulation has only one design variable for each arc of the graph. Consider the binary variables x_{ij} indicating whether or not arc (i, j) is in the solution as well as the network flow binary variables y_{ij}^{hk} used in the previous models. After some simplifications (for simplicity we skip the details here) the Hop-MCF formulation can be rewritten as follows

Model Hop-MCF

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{38}$$

$$\text{s.t.} \quad \sum_{i\in V\setminus\{j\}} x_{ij} = 1, \qquad j \in V \setminus \{0\} \tag{39}$$

$$(7), (8), (10), (12), (13), (14)$$

$$y_{0j}^{1k} \leq x_{0j}, \qquad (0, j) \in A; k \in V \setminus \{0\} \tag{40}$$

$$\sum_{h=2}^{H} y_{ij}^{hk} \leq x_{ij}, \qquad (i, j) \in A; i \neq 0; k \in V \setminus \{0\}; \tag{41}$$

$$x^{ij} \in \{0, 1\}, \qquad (i, j) \in A \tag{42}$$

$$\tag{43}$$

4.2 Comparing the Revised Hop-NF and Hop-MCF Models

The Hop-MCF rewritten in this form permits us to compare quite easily its linear programming relaxation with the linear programming relaxation of the revised Hop-NF model. In fact, by noting that any feasible for the revised Hop-NF model can be transformed into a feasible solution of the Hop-MCF formulation by using the linear equalities

$$x_{0j} = X_{0j}^{1}, \qquad \forall j \in V \setminus \{0\} \tag{44}$$

$$x_{ij} = \sum_{h=2}^{H} X_{ij}^{h}, \qquad \forall (i, j) \in A; i \neq 0 \tag{45}$$

we obtain

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

Proposition 1 *The projection of the set of feasible solutions of the revised Hop-NF model augmented with the equalities (44, 45) into the space of the* (x_{ij}, y_{ij}^{hk}) *variables is contained in the set of feasible solutions of the linear programming relaxation of the Hop-MCF model.*

That is, under (44, 45) the only essential difference between the two models are constraints (17) in Hop-NF which are a disaggregated version of constraints (41) in Hop-MCF (later on, with a small example we will give some intuition on why these two sets have different modelling properties). Since adding equalities (44, 45) to Hop-NF does not modify its linear programming value, we conclude that

Proposition 2 *The linear programming relaxation value of the revised Hop-NF model is at least as good as the linear programming relaxation value of the Hop-MCF model.*

Before discussing whether the inequality in Proposition 2 is strict, we note and emphasize that computational results reported in [14, 19], and [8] show that the linear programming bound of this formulation is in many cases equal to the integer optimal value for several medium sized (up to 40 nodes) instances tested. [8] compare several alternate and theoretically equivalent approaches to compute this linear programming bound. The general idea one gets from the results on those papers is that these approaches have difficulties on instances with 80 or more nodes.

Proposition 2 implies that we can do at least as good with the new models proposed in this paper. However, we may not gain much by using the new network flow models discussed in Section 2 since these models may suffer from the same drawbacks as the previous Hop-MCF model. Fortunately, the observation that the HMSTP is nothing else than a special Steiner tree problem in the layered graph will allows us to use the method described in Section 5 to optimize over the equivalent Hop-Cut model.

We conclude this subsection by observing that, for some instances, the linear programming bound of the new models can be strictly better than the linear programming bound of Hop-MCF. The solution shown in Figure 3 is an extreme point of the feasible set of the linear programming relaxation of Hop-MCF, for an instance with $n = 4$ and $H = 3$. On the left-hand side it depicts the x_{ij} variables with value 0.5. The corresponding y_{ij}^{hk} variables for $k = 3$ and 4, also with value 0.5, are depicted on the right-hand side of the same figure.

This solution is not feasible for the Hop-NF model. To see this, note that the value of the variables y_{24}^{23}, y_{24}^{34} are equal to 0.5 in the solution of the linear programming relaxation of the Hop-MCF formulation. However, the constraints 33 from the new model would imply $X_{24}^{2} = X_{24}^{3} = 0.5$ which is infeasible in the presence of the constraints 45 $X_{24}^{2} + X_{24}^{3} = x_{24}$ linking the two models. This solution also gives some intuition why the new models have a stronger linear programming relaxation. In the previous model Hop-MCF, a "fractional" arc (like arc $(2, 4)$) is allowed to be in different positions in different paths (position

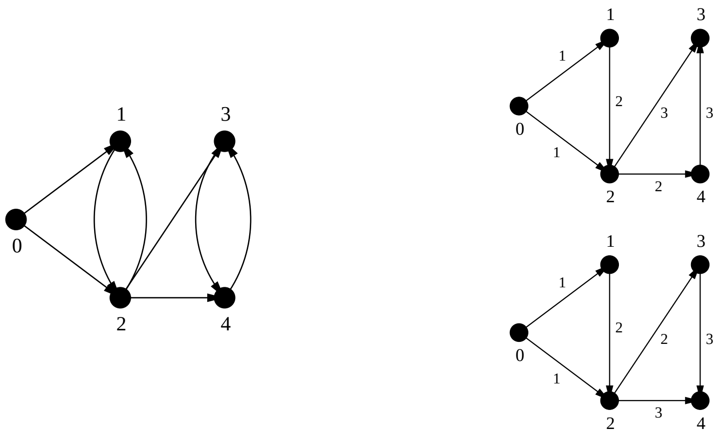

Figure 3: Fractional solution over the x variables (on the left-hand side) and corresponding y variables for $k = 3$ and 4 (on the right-hand side), labels in the arcs indicate hop index h

3 in the path to node 4 and in position 2 in the path to node 3). A similar situation happens with respect to arc (2,3). These situations are not allowed in the new models.

4.3 The Jump Formulation

The formulation proposed in this subsection has been proposed in [6] and it has been tested in [8]. In order to describe the formulation we need to introduce some concepts. Let $S_0, S_1, \ldots, S_{H+1}$ be node-disjoint sets defining a partition of the whole node set V such that each subset is nonempty, and that $0 \in S_0$. We then define $J = J(S_0, S_1, \ldots, S_{H+1}) = \cup_{[i+1<j]}[S_i, S_j]$ where $[S_i, S_j]$ is the set of arcs $\{(u, v) \in A : u \in S_i, v \in S_j\}$. We call such set J a H-jump. Let J_H denote the set of all H-jumps. Consider, now, the following set of inequalities, the jump inequalities,

$$\sum_{(p,q)\in J} x^k_{pq} \geq 1, \quad J \in J_H \tag{46}$$

To see that (46) are valid for any H we consider the following argument that is a simple adaptation from the one given in [5] for hop-constrained paths. Assume we place all the nodes in $H + 2$ consecutive layers where the i^{th} layer corresponds to the nodes in S_i. Assume also that a path with at most H hops from the root node to any node $k \in S_{H+1}$ does not contain any arc in a given jump J. Then, the path would contain one arc in each of the (pairwise disjoint) arc sets $[S_i, S_{i+1}]$ and then it would contain at least $H + 1$ arcs which is a contradiction. Thus, any such path must have at least one arc in each jump J. Loosely speaking, any feasible needs to make a jump somewhere from a node set S_i to one of the sets $S_{i+2}, \ldots, S_{H+1}$. [6] proved that these inequalities are facet defining when $|S_0| = |S_{H+1}| = 1$, for all $H \geq 3$.

The formulation Jump is as follows

Formulation Jump

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{47}$$

$$\text{s.t.} \quad (39), (42)$$

$$\sum_{(p,q)\in[V\setminus S,S]} x_{pq} \geq 1, \quad S \subseteq V; 0 \notin S \tag{48}$$

$$\sum_{(p,q)\in J} x_{pq} \geq 1, \qquad k \in V \setminus \{0\}; J \in J_H \tag{49}$$

Note that constraints (48) are the usual cut constraints and constraints (49) are the jump inequalities for all terminal nodes. [8] have shown that the projection of the set of feasible solutions of the linear programming relaxation of Hop-MCF into the space of the x_{ij} variables is equal to the set of feasible solutions of the linear programming relaxation of the Jump model when $H = 3$ and that strict inclusion occurs when $H > 3$. They also showed that a characterization of the linear programming relaxation of Hop-MCF in the space of the x_{ij} variables is rather complicated when $H = 4$ (this follows from a result given in [7] where the authors show that a complete description of the dominant of hop constrained paths contains exponential sized sets of constraints of the form

$$\sum_{(i,j)\in C} a_{ij}y_{ij} \geq r \tag{50}$$

for adequate arc sets C and where r and a_{ij} are positive integers - for more details, the reader is referred to [8]).

As suggested in [8], albeit having a LP relaxation which is weaker than the LP relaxation of the Hop-MCF formulation for $H > 3$, the Jump formulation may be worth investigating from a computational point of view because it contains only one variable for each arc in the input graph and the memory drawbacks encountered with the Hop-MCF model could be overcome. Since the convex hull of spanning trees without hop constraints rooted at node 0 is completely described by (39), (48) and nonnegativity constraint on the x_{ij} variables, a Lagrangean relaxation scheme where the relaxed problem is a directed spanning tree and the constraints being penalized are the jump constraints (49), is worth investigating. Such a method was tested in [6]. As the relaxed solutions are integer, separation of the jump inequalities can be done in polynomial time and since there are too many jump constraints, an implicit generation scheme was implemented. Unfortunately, the results reported in that paper were very disappointing. The main explanation for this bad behavior appears to be the fact that it is not yet clear what is the best strategy for separating the jump inequalities, that are violated in a current iteration of the method, to be added to the Lagrangean dual.

4.4 New Inequalities

As noted before, the linear programming bounds of the new formulations are strictly better than the linear programming bounds of the Hop-MCF formulation, even for $H = 3$. The result given in [8] suggests that even for that case, it would be interesting to have some insight of what inequalities are implied by the linear programming relaxation of the new formulation and that are not redundant in the linear programming relaxation of the Jump formulation.

We show next that several sets of such inequalities correspond to stronger versions of inequalities that result either from adding several jump inequalities or by adding jump inequalities with cut inequalities. More precisely, we will show that the linear programming relaxation of the Hop-Cut model implies inequalities of the form:

$$\sum_{k=1}^{K_j} x(J_k) + \sum_{k=1}^{K_c} x(C_k) \geq |K_j| + |K_c| + \sum_{i,j} \gamma_{ij} x_{ij}, \tag{51}$$

where J_k, $k = 1 \dots K_j$, are jump sets, C_k, $k = 1 \dots K_c$, are cut sets and γ_{ij} are nonnegative lifting coefficients.

The intuition behind this statement is based on the following two observations. First, it is easy to show that the linear programming relaxation of the Hop-Cut formulation implies the following lifted versions (in the x_{ij} and X_{ij}^h space) of the jump and the cut inequalities (given in a generic form):

$$\textstyle\sum_{(p,q)\in J} x_{pq} \geq 1 + \sum_{i,j,h} \alpha_{ij}^h X_{ij}^h, \tag{52}$$

$$\textstyle\sum_{(p,q)\in C} x_{pq} \geq 1 + \sum_{i,j,h} \beta_{ij}^h X_{ij}^h, \tag{53}$$

where J and C are jump and cut sets respectively, and α_{ij}^h and β_{ij}^h are nonnegative lifting coefficients.

Second, these two sets of lifted inequalities suggest that if we add several constraints 52 or add some constraints 52 with some constraints 53, we might be able to use the linking constraints 44 and 45 on some of the variables on the right side of the resulting inequality in order to obtain a constraint of the form 51. As an example, suppose that for a given arc (i, j), $i \neq 0$, cut C and jump J we are able to obtain the following two lifted inequalities

$$\textstyle\sum_{(p,q)\in J} x_{pq} \geq 1 + \sum_{h\in H_1} X_{ij}^h, \tag{54}$$

$$\textstyle\sum_{(p,q)\in C} x_{pq} \geq 1 + \sum_{h\in H_2} X_{ij}^h, \tag{55}$$

where $H_1 \cup H_2 = \{2, ..., H\}$. Adding up the two inequalities and using 45 gives:

$$\textstyle\sum_{(p,q)\in J} x_{pq} + \sum_{(p,q)\in C} x_{pq} \geq 2 + x_{ij}, \tag{56}$$

which is stronger than the the inequality obtained by adding the original jump and cut inequality. Clearly, similar examples with more arcs on the right-hand side may also hold.

4.4.1 Generating Lifted Jump and Lifted Cut inequalities in the space of the x_{ij} and X_{ij}^h variables

First, we show how to obtain one general version of a lifted jump inequality 52. For a fixed hop parameter H, let us consider a partition $S_0, S_1, \ldots, S_{H+1}$ with $0 \in S_0$ and let us considered the layered graph partitioned in a similar manner, that is, set S_{ih} corresponds to the node set S_i in the layer h of the layered graph (for $i = 1, \ldots, H+1$ and $h = 1, \ldots, H$). Now, consider a cut $[A, B]$ in the layered graph such that

$$B = \bigcup_{h=1,\ldots,H}(\bigcup_{i=h+1,\ldots,H+1} S_i^h). \tag{57}$$

Figure 4 gives an example of such a cut in the layered graph for an instance with $H = 3$ and $n = 4$. Now, by adding adequate trivial inequalities $X_{ij}^h \geq 0$ to the left-hand side of the cut inequality and by using the linking constraints 44 and 45 we obtain the following lifted jump inequality

$$\sum_{(p,q)\in J} x_{pq} \geq 1 + \sum_{j=1}^{H-2}\sum_{i=j+2}^{H}\sum_{h=i}^{H} X^h(S_j, S_i) + \sum_{j=2}^{H-1}\sum_{i=j+2}^{H+1}\sum_{h=2}^{j} X^h(S_j, S_i),$$

$$J = J(S_0, S_1, \ldots, S_{H+1}) \tag{58}$$

Note that the X_{ij}^h variables used on the expression on the righthand side of the inequality correspond to the trivial inequalities that have been added to the original cut inequality. Note also, that the arcs associated to these variables either are inside the set A (arcs associated to variables in the first summation term in the right hand-side of the inequality) or inside the set B (arcs associated to variables in the second summation in right hand-side) of the cut.

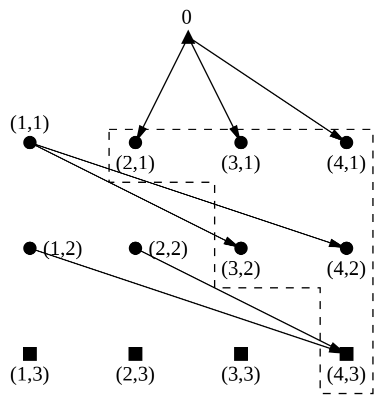

Figure 4: Cut in the layered graph corresponding to a lifting of a jump inequality in the original graph

As an example consider an instance with $H = 3$ and $n = 4$, and the partition

$J = J(\{0\},\{1\},\{2\},\{3\},\{4\})$. Then, the previous inequality becomes

$$\sum_{(p,q)\in J} x_{pq} = x_{02} + x_{03} + x_{04} + x_{13} + x_{14} + x_{24} \geq 1 + X^3_{13} + X^2_{24}, \qquad (59)$$

as shown in Figure 4.

With respect to the lifted cut inequality 53, let S, $1 < |S| < n$ be a set in the original graph not containing the root and consider the cut set $C = [V \setminus S, S]$. For any vertex $k \in S$, the lifted cut inequality is as follows:

$$\sum_{(p,q)\in C} x_{pq} \geq 1 + \sum_{i\in V\setminus S, i\neq 0} \sum_{j\in S\setminus\{k\}} X^H_{ij}. \qquad (60)$$

For this inequality we have omitted the derivation. However, a simple example shows how to obtain the corresponding derivation. Consider, again, $n = 4$ and $H = 3$. When $S = \{3, 4\}$ and $k = 3$, we obtain

$$\sum_{(p,q)\in C} x_{pq} = x_{03} + x_{04} + x_{13} + x_{14} + x_{23} + x_{24} \geq 1 + X^3_{14} + X^3_{24}, \qquad (61)$$

as shown in Figure 5.

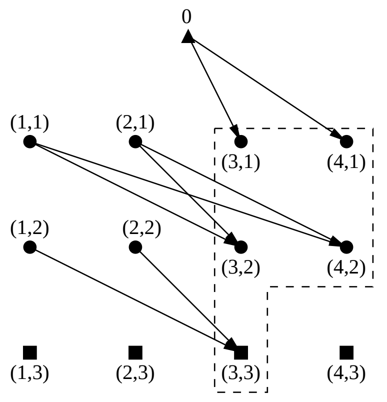

Figure 5: Cut in the layered graph corresponding to a lifting of a cut inequality in the original graph

4.4.2 Generating new inequalities in the x_{ij} space

Fist, we show how to generate a constraint that can be seen as a stronger version of an inequality that is obtained by adding a lifted jump inequality 58 with a lifted cut inequality 60. The main observation to generate the new inequality is to note that in the lifted variables of 60 the hop index h is always equal to H and that the second summation in the right hand side of 58 contains the expression

$$\sum_{h=2}^{H-1} X^h[S_{H-1}, S_{H+1}].$$

Thus, if for a fixed hop parameter H, we consider a partition $J = \{S_0, S_1, \ldots, S_{H+1}\}$ with $S_0 = \{0\}$, for the lifted jump inequality and, for the lifted cut inequality, we consider a set S such that S_{H+1} is contained in S and S_{H-1} is not contained in S, and add the two inequalities, we obtain

$$\textstyle\sum_{(p,q)\in J} x_{pq} + \sum_{(p,q)\in C} x_{pq} \geq 2 + \sum_{h=2}^{H} X_h[S_{H-1}, S_{H+1}] + \text{“extra terms”} \quad (62)$$

By discarding the extra terms and using 45 on the summation, we obtain the *cut/jump inequality*:

$$\textstyle\sum_{(p,q)\in J} x_{pq} + \sum_{(p,q)\in C} x_{pq} \geq 2 + x[S_{H-1}, S_{H+1}]. \quad (63)$$

For an instance with $n = 4$, $H = 3$, $J = J[\{0\}, \{1\}, \{2\}, \{3\}, \{4\}]$, C=[{0,1,2}, {3,4}] the previous expression becomes

$$x_{02} + 2x_{03} + 2x_{04} + 2x_{13} + 2x_{14} + x_{24} + x_{23} \geq 2, \quad (64)$$

which is exactly the inequality obtained from the sum of the two lifted inequalities 59 and 61. It can be seen that this inequality cuts the fractional solution shown in Figure 3. For instances with $H = 4$ and 5, we have also found instances whose optimal linear programming solutions of the Hop-MCF model do not satisfy constraints 63.

The second inequality is obtained by combining several jump inequalities obtained in a “circular” fashion. Consider a partition of the node set $V \setminus \{0\}$ into $H + 1$ subsets $A_1, A_2, \ldots, A_H, A_{H+1}$ and consider $H + 1$ lifted jump inequalities 58 where the subsets are as follows. First inequality: $S_0 = \{0\}, S_1 = A_1, S_2 = A_2, \ldots, S_H = A_H, S_{H+1} = A_{H+1}$; second inequality: $S_0 = \{0\}, S_1 = A_2, S_2 = A_3, \ldots, S_H = A_{H+1}, S_{H+1} = A_1$; ...; H-th inequality: $S_0 = \{0\}, S_1 = A_H, S_2 = A_{H+1}, \ldots, S_H = A_1, S_{H+1} = A_2$; and $(H + 1)$-th inequality $S_0 = \{0\}, S_1 = A_{H+1}, S_2 = A_H, \ldots, S_H = A_2, S_{H+1} = A_1$. After adding all of these inequalities, using 44 and 45 on the right-hand side and discarding some variables, we obtain the following inequality (for simplicity we omit the derivation here):

$$Hx[\{0\}, V \setminus \{0\}] + \sum_{i=1}^{H+1} \sum_{\substack{j = 1; j \neq i, \\ (j-1) \neq i \bmod H}}^{H+1} x[S_i, S_j] \geq H + 1. \quad (65)$$

For example, if $n = 4$, $H = 3$, $S_0 = \{0\}, S_1 = \{1\}, S_2 = \{2\}, S_3 = \{3\}, S_4 = \{4\}$, we obtain:

$$3x_{01} + 3x_{02} + 3x_{03} + 3x_{04} + x_{13} + x_{14} + x_{21} + x_{24} + x_{31} + x_{32} + x_{42} + x_{43} \geq 4 \quad (66)$$

Again, experiments with small instances have shown that these inequalities are not redundant in the linear programming relaxation of the Hop-MCF formulation.

By presenting these two sets of inequalities we have given some intuition (in the design space) of what gain by modelling the HMSTP in the layered graph.

5 The Branch-and-Cut Algorithm

The branch-and-cut algorithm used to solve model Hop-Cut is based on the algorithm proposed by [31] for the STP. The main improvement of this algorithm over previous algorithms based on the same directed cut model, such as the one presented in [24], lies in the effective use of dual and primal heuristics to fix variables and speed up the convergence of the cutting plane generation. The following algorithmic elements were incorporated into the code used to solve model Hop-Cut:

1. **Dual Ascent Heuristic:** Initially proposed by [38] for the directed multi-commodity flow model, it can be straightforwardly adapted for the directed cut model. This fast heuristic usually yields quite good lower bounds, typically less than 5% bellow the optimal value.

2. **Hot-Starting the Cut Generation:** The Dual Ascent Heuristic may be used to provide a good initial set of cuts to be used in the cutting plane generation. In fact, by solving the linear program with those cuts one always obtain a lower bound at least as good as the one provided by Dual Ascent Heuristic. This hot-start may greatly reduce the number of cut rounds required to solve the linear programming relaxation of the model.

3. **Using the Fractional Solutions to Guide Primal Heuristics:** An initial primal solution is obtained by applying the well-known Prim Shortest Path Heuristic (Prim-SPH) (see [34]) over the layered graph and improving it by the key-path and node local search [32]. The solutions obtained in this way are only reasonably good. However, after each iteration of the cutting plane method, the fractional solutions of the linear programs are used to compute pseudo-costs, that are, then, feed to the Prim-SPH. Those pseudo-cost give incentives for using arcs that appear consistently in those fractional solutions. The subsequent local search is then applied using the original costs.

4. **Fixing by Reduced Costs:** The value of a known primal solution permits to remove variables from the model by using reduction tests based on arc reduced costs provided either by the Dual Ascent Heuristic or by subsequent linear programs. The removed variables would never appear in any improving solution. The fixing by reduced costs can be very effective when both primal and dual solutions are close to the optimal.

More details about the above mentioned elements can be found in [31, 35, 37].

6 The Diameter Constrained Spanning Tree Problem

In this section we adapt the previous methods to a variation of the HMSTP, the Diameter constrained Spanning Tree problem (DMSTP). Given a prescribed graph $G = (V, E)$ with node set V and edge set E as well as a positive cost c_e associated with each edge e of E, we wish to find a minimal spanning tree with a bound D on the diameter of the tree, which is the maximum number of edges in any of its paths. When $D = 2$ or 3, the problem is easy to solve. However, it is NP-Hard when $D \geq 4$ (see [9]). As noted before, the DMSTP differs from the HMSTP in the sense that here we constrain the path between each pair of nodes while in the HMSTP, only the paths from the special node are constrained. This observation suggests that the DMSTP appears to be much more complex than the HMSTP.

However, several approaches for the DMSTP (see, for instance, [1, 16, 33, 20]) have used the properties of tree centers in order to transform the DMSTP into special versions of the HMSTP. For instance, with respect to situations with parameter D even the, following center property proves to be very useful.

Property 1 *A tree T has diameter no more than an even integer D if and only if some node p of T satisfies the property that the path from node p to any other node of the tree contains at most $D/2$ edges.*

This property permits us to transform the DMSTP, for situations when D is even, into an HMSTP in an enlarged graph with an extra root node, node 0, and which is connected to any other node by an edge of zero cost. In the enlarged graph, one must determine a hop constrained spanning tree with depths at most $D/2 + 1$ and such that the degree of the extra node is equal to 1 (the edge emanating from this node will determine the node of the original graph that will serve as the center of the diameter constrained tree). The extra degree constraint is easy to incorporate in the code described in the previous section. Thus, the complexity of solving the DMSTP with diameter D even is essentially the same as solving the HMSTP.

In our computations we use the model described in Section 3 with the small adaptation mentioned in the previous paragraph. An example depicting the transformation is shown in figures 6 and 7. Since, the model in [16] is the same as the model in [14], we can use Proposition 2 to say that for D even, the linear programming relaxation of the adapted layered graph model we use here is at least as good as the linear programming relaxation of the model proposed by [16]. Again, by using our exposition in Section 4, it is not difficult to argue that for some instances, strict dominance occurs.

For situations when D is odd, there appears not to be any agreement in what should be the best model to use. The reason for this is that the straightforward adaptation of the center property for D odd ("a tree T has diameter no more than D if and only if some edge (p, q) of T satisfies the property that the path from any other node of the tree to p or q contains at most $(D - 1)/2$

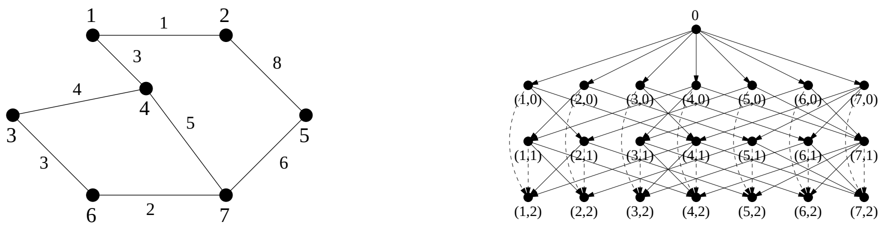

Figure 6: Transformation of DMSTP for D even: original graph of an instance with $D = 4$ on the left-hand side and the corresponding layered graph on the right-hand side.

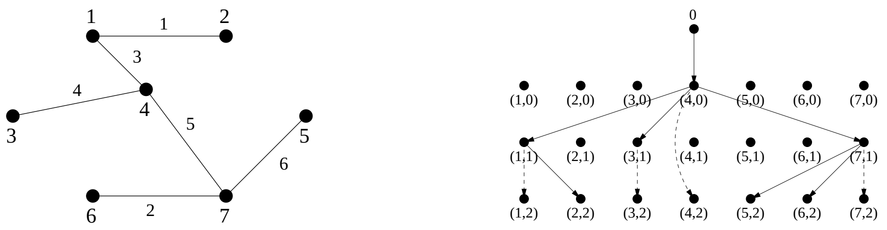

Figure 7: Optimal DMST in the original graph (cost 21) and its corresponding Steiner tree in G_L ($D = 4$)

edges") is not easy to combine either with a Network Flow model (as in [14]) or with the layered graph approach proposed here unless one allows the creation of special nodes corresponding to the edges of the graph (as suggested and tested in [16]. However, this modification leads to oversized networks and the proposed approaches on these networks usually take much more time (when computer storage requirements do not become a bigger problem) than similar approaches used for situations with D even. Alternative approaches have been suggested in [16] and [15] where 2-path (or 2-flow) and 2-tree based models where efficiently combined with 2-center properties for trees with diameter odd. However, these properties do not appear to be easily combined with the layered graph approach proposed in the previous sections.

A different and perhaps less straightforward transformation for the situations when D is odd and which is suitable to be modelled by the layered graph approach is described as follows: add two special dummy layers (0 and -1) corresponding to the possible choices of nodes to be the extremes of the central edge of the tree. Node 0 is connected to all nodes in layer 0 with zero cost arcs. We add a degree constraint guaranteeing that exactly one of these edges is chosen. Every node in layer 0, $(i, 0)$, is linked to any node $(j, -1)$, such that $(i, j) \in E$, in layer -1 by an arc of cost c_{ij}. We also add a constraint stating that exactly one of these edges is chosen. This constraint and the previously described constraint guarantee that the two extremes of the central edge are chosen and its cost is accounted. Then we add arcs of cost 0 linking each pair of nodes $(i, -1)$ and $(i, 0)$. In this way we guarantee that the two nodes

$(i, 0)$ and $(j, 0)$ are selected if edge (i, j) is the central edge of the tree. The remainder of the layered of the graph is as before. Figure 8 shows a graph and the corresponding layered graph for $D = 5$ and Figure 9 gives an example of corresponding solution in the two graphs.

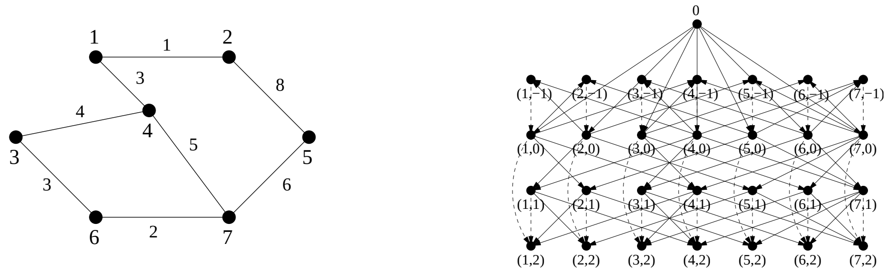

Figure 8: Transformation of DMSTP for D odd: original graph of an instance with $D = 5$ on the left-hand side and the corresponding layered graph on the right-hand side.

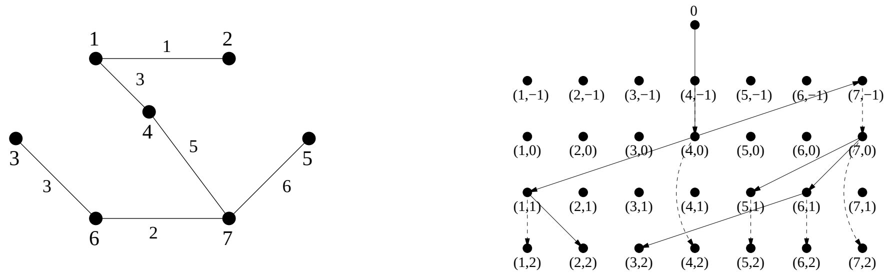

Figure 9: Optimal DMST in the original graph (cost 20) and its corresponding Steiner tree in G_L ($D = 5$)

We note that it does not appear to be easy to relate the new model for D odd with previous modelling approaches since the transformations appear to be quite different.

7 Computational Results

In this section we give results with the branch-and-cut method described in section 5 for the two problems, the HMSTP and the DMSTP. The tests were performed in a PC Intel Core 2 Duo, 2.2 GHz, with 2Gb of RAM. The method was implemented using the callable routines in XPRESS-MP 2007A, that both solves linear programs and may perform branch to obtain the optimal integer solutions.

7.1 Computational Results for the HMSTP

The experiments use some complete graph instances on 21, 41, 61, 81, 101, 121 and 161 nodes that have been used in previous papers for the HMSTP (see, for instance, [8]) and for other constrained spanning trees, namely the capacitated minimum spanning tree problem (see, for instance [36]). For each size, we have considered one random cost instance, denoted by TR, and two Euclidean cost instances. Two locations for the root are considered for the Euclidean instances: Instances that have the root located in the center of the grid, denoted by TC, and instances that have the root located on a corner of the grid, denoted by TE. The hop parameter H was set to 3, 4 and 5 in every case.

In order to reduce the size of each instance, we have used the following simple arc elimination test (see [12]). If $c_{ij} > c_{0j}$, then any optimal solution does not uses arc (i, j) and if $c_{ij} = c_{0j}$ $(i \neq 0)$, then there is an optimal solution without arc (i, j). This means that arc (i, j) can be eliminated whenever $c_{ij} \geq c_{0j}$. This arc elimination test is applied to every instance before its solution. Table 1 shows, for each instance, the percentage of number of arcs still remaining after the elimination test was performed.

Tables 2 to 4 give information on the performance of the method for the TC, TR and TE instances. The first column gives the number of nodes (without the root node) of the corresponding graph. The second column gives the value of H. The third column gives the value of the integer optimal value and the fourth gives the value of the bound produced by the Hop-Cut model. The next columns give, respectively, the value of the lower bound obtained by Dual Ascent Heuristic, the value of the upper bound obtained by the first application of Prim-SPH + local search and the number of inequalities (3) added in the branch-and-cut algorithm. Finally, the last column gives the total CPU time in seconds spent by the whole method (a time of 0 indicates that the instance was solved in less than 1 milisecond, usually because both Dual Ascent and Prim-SPH found the same bound). In all instances, except on instance TE160 with $H = 5$, the solution of the linear programming relaxation of the Hop-Cut model was integral and no branching was performed.

Table 5 compares the new method, in terms of bounds and CPU times (see column LP-HOP-CUT and subsequent column), with the lower bounds taken from the model of [14] described in Section 4 (see column LP-G and subsequent column), on instances with $n = 40$ and $n = 80$. These results show that the theoretical dominance of the new lower bound is reflected in practice. They also show that the new approach (namely, the branch-and-cut method applied to the theoretical strong Hop-Cut model) solves quite easily instances with up to 160 nodes. Previously, only some instances with up to 80 nodes have been solved. The computational results of the model of [14] were obtained on a PC Pentium IV, 2.4GHz with 768Mb of RAM and used the callable routine CPLEX 7.1.

n	20	40	60	80	100	120	160
TC	26%	27%	25%	25%	25%	27%	53%
TE	70%	67%	70%	68%	70%	75%	78%
TR	44%	46%	51%	46%			

Table 1: Remaining size of reduced HMSTP instances.

n	H	OPT	LP	DA	PRIM-HOP	NCUTS	T(s)
	3	340	340	340	340	-	0
20	4	318	318	318	318	-	0
	5	312	312	312	312	-	0
	3	609	609	601	609	190	0.06
40	4	548	548	540	548	213	0.07
	5	522	522	516	524	337	0.24
	3	866	866	857	892	479	0.35
60	4	781	781	775	795	834	1.26
	5	734	734	732	734	153	0.15
	3	1072	1072	1066	1084	556	0.49
80	4	981	981	973	995	2028	46.9
	5	922	922	920	934	1286	6.57
	3	1259	1259	1237	1290	1843	15.3
100	4	1166	1166	1158	1198	3206	190
	5	1104	1104	1098	1116	3590	251
	3	1059	1059	1051	1102	1186	2.68
120	4	926	926	921	962	1743	25.9
	5	853	853	849	875	2369	103
	3	1357	1357	1349	1426	2683	160
160	4	1133	1133	1130	1163	3495	643
	5	1039	1039	1033	1055	8431	10292

Table 2: TC instances.

7.2 Computational Results for the DMSTP

For the computational experiments concerning the DMSTP we have performed two groups of tests. In the first group, our aim is to compare the method described in this paper with the other methods described in the recent literature. In the second group we consider much larger instances in order to show that the method proposed in this paper can solve many instances that can not be solved previous methods.

For the first group we report results given by the methods described in [33], [20], [16] (for situations with D even), [17] (for situations with D odd), and [29]. The first method uses formulations based on enhanced versions of the Miller-Tucker-Zemlin constraints. The second uses formulations based on precedence variables. The method described in [16] is based in the Hop-MCF formulation readapted for the DMSTP as explained in section 6.2. The method

n	H	OPT	LP	DA	PRIM-HOP	NCUTS	T(s)
	3	168	168	168	168	-	0
20	4	146	146	146	146	-	0
	5	137	137	137	137	-	0
	3	176	176	172	177	126	0.05
40	4	149	149	147	149	154	0.13
	5	139	139	139	139	-	0
	3	213	213	206	217	316	0.17
60	4	152	152	151	153	240	0.18
	5	124	124	124	124	-	0.02
	3	208	208	206	208	170	0.14
80	4	180	180	178	180	267	0.36
	5	164	164	164	164	-	0.07

Table 3: TR instances.

n	H	OPT	LP	DA	PRIM-HOP	NCUTS	T(s)
	3	449	449	449	457	114	0.03
20	4	385	385	385	385	-	0.01
	5	366	366	361	366	122	0.05
	3	708	708	708	728	301	0.13
40	4	627	627	624	629	267	0.14
	5	590	590	589	596	398	0.41
	3	1525	1525	1521	1569	488	0.48
60	4	1336	1336	1328	1373	923	3.43
	5	1225	1225	1225	1229	374	0.46
	3	1806	1806	1802	1840	753	1.24
80	4	1558	1558	1549	1580	917	4.29
	5	1442	1442	1435	1477	2878	226
	3	2092	2092	2082	2111	1344	9.85
100	4	1788	1788	1771	1888	2887	356
	5	1625	1625	1625	1699	1592	36.6
	3	1267	1267	1258	1352	1708	15.6
120	4	1074	1074	1071	1155	3043	574
	5	969	969	962	1000	5071	3397
	3	1496	1496	1488	1616	2282	76.2
160	4	1229	1229	1221	1286	6458	5626
	5	1107	1106.5	1098	1182	15764	53797

Table 4: TE instances.

described in [17] is based on two center properties of D diameter situations. The more recent method [29] is based on constraint programming. These instances are taken from [33] and include dense graph instances with up to 40 nodes. Although the methods described in [16] and [17] have been originally tested for

PROB, n	H	OPT	LP_HOP-CUT	T(s)	LP_G	T(s)
	3	609	609	0.06	604.5	2+48
TC, 40	4	548	548	0.07	547	8+3
	5	522	522	0.24	522	13
	3	176	176	0.05	176	2
TR, 40	4	149	149	0.13	148.3	9+3
	5	139	139	0	139	26+1
	3	708	708	0.13	701.7	175+1984
TE, 40	4	627	627	0.14	625.3	969+9797
	5	590	590	0.41	588.1	2794+23052
	3	1072	1072	0.49	1069	157+3204
TC, 80	4	981	981	46.9	976.5	1811+23659
	5	922	922	6.57	920.6	4198+7590
	3	208	208	0.14	208	64+3
TR, 80	4	180	180	0.36	180	676+4
	5	164	164	0.07	164	1271+6
	3	1806	1806	1.24	1792.5	16127+(r)
TE, 80	4	1558	1558	4.29	1544.5	160127+(r)
	5	1442	1442	226	-	-

Table 5: Comparing with the Hop-MCF model.

sparse graph instances, we have specially run them (on a Pentium IV 2.4 GHz) for these dense graph instances in order to permit us to compare all methods in the same data set. Table 7.2 shows for each method, the percent root gap (except for the constraint programming method) and the total time in seconds to solve the instance. Times from [33] and [20] were both obtained on a Pentium IV 2.8 GHz, those from [29] on a Pentium IV 3 GHz. The last row in the table show averages.

For the second group we use the same instances as the ones used for the HMSTP, with up to 161 nodes. In these results one can, again, see that the proposed method is quite strong in the sense that for almost all tested instances the linear programming bound is optimal. It is also curious to see the effect of the new transformation for situations when D is odd. In our results, it does not appear to be any significant difference between the two cases, D even and D odd, which is in contrast with previous transformations / approaches. Figure 10 depicts an optimal solution of the TE instance with 161 vertices, for $D = 5$.

References

[1] N. R. Achuthan, L. Caccetta, P. A. Caccetta, and J. F. Geelen. Computational methods for the diameter restricted minimum weight spanning tree problem. *Australasian Journal of Combinatorics*, 10:51–71, 1994.

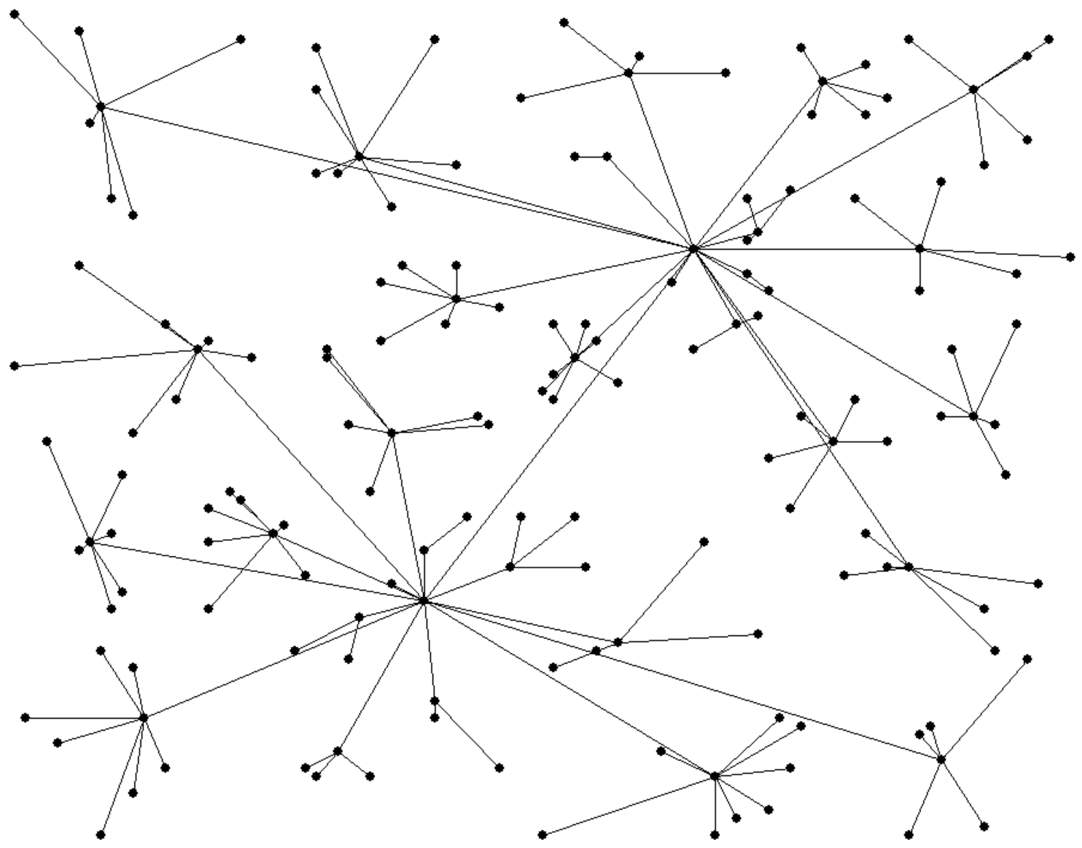

Figure 10: Optimal solution of DMSTP instance te161 with $D = 5$.

[2] S. Chopra and M. R. Rao. The Steiner tree problem I: Formulations, compositions and extension of facets. *Mathematical Programming*, 64(2):209–229, 1994.

[3] A. M. Costa, J-F. Cordeau, and G. Laporte. Models and branch-and-cut algorithms for the steiner tree problem with revenues, budget and hop constraints. Technical Report CRT- 2006/14, Centre de Recherche sur les Transports, Montreal, 2006.

[4] G. Dahl. The 2-hop spanning tree problem. *Operations Research Letters*, 23:21–26, 1998.

[5] G. Dahl. Notes on polyhedra associated with hop-constrained paths. *Operations Research Letters*, 25:97–100, 1999.

[6] G. Dahl, T. Flatbert, N. Foldnes, and L. Gouveia. The jump formulation for the hop-constrained minimum spanning tree problem. Technical Report CIO 5, University of Lisbon, 2004.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

				Hopcut		LiftDMST		ILP		GMR04/GM03		CP
$\|V\|$	$\|E\|$	D	OPT	Gap	T(s)	Gap	T(s)	Gap	T(s)	Gap	T(s)	T(s)
15	105	4	346	0	0.05	23.9	4.7	22	0.7	0.9	3.09	0.08
		5	331	0	0.06	28.9	22.8	29.2	3	0.5	5.7	0.22
		6	314	0	0.06	16.4	18.6	32.8	8.1	0.5	163.69	0.28
		7	303	0	0.07	10.6	26.9	10.6	20	0.0	6.23	0.38
		9	290	0	0.07	6.6	6.2	6.6	10.7	0.0	7.03	0.47
		10	286	0	0.08	8.3	1	5.3	4.3	0.0	10.59	0.41
20	190	4	349	0	0.07	24.4	562.9	25	2.5	0.1	11.97	0.2
		5	414	0	0.08	20.9	436.7	17.8	8.1	1.1	173.59	1.06
		6	298	0	0.06	12.5	455.2	21.7	95	2.2	1590.03	2.03
		7	333	0	0.01	7.7	5.1	5	4.5	0.1	13.17	0.97
		9	327	0	0.01	8.9	73.4	8.8	66.7	0.0	27.31	5.01
		10	324	0	0.01	8.1	29.7	8.1	101	0.0	41.25	6.08
25	300	4	500	0	0	26.6	15203.7	23.2	12	0.0	55.59	1.48
		5	429	0	0.07	22.6	>20000	23.1	64.3	0.3	55.81	2.83
		6	378	0	0.11	11.4	826.7	17	26.4	0.0	193.67	39.14
		7	408	0	0.13	15.9	11521.3	17.6	770.5	1.1	2647.51	56.06
		9	336	0	0.11	8	246	5.2	295.4	0.0	103.27	114.14
		10	379	0	0.11	5.9	254.8	5.3	404.9	0.0	210.2	55.47
20	50	4	442	0	0.03	16.5	1	13.4	0.2	0.2	0.56	0.05
		5	381	0	0	57.8	4.6	58.8	1	0.0	0.52	0.17
		6	329	0	0	9.5	0.8	11.6	5.1	0.0	0.91	0.13
		7	362	0	0.06	8.4	0.8	25.5	1.2	0.0	0.75	0.14
		9	362	0	0.01	7.7	0.7	10.4	2.8	0.0	0.59	0.45
		10	359	0	0	3.9	0.2	2.2	1.3	0.0	2.28	0.64
40	100	4	755	0	0.07	27.7	43.7	19.4	1.9	0.0	1.45	5.44
		5	729	0	0.08	52.9	291.7	47.5	6.4	0.0	2.39	7.31
		6	599	0	0	4.6	50.9	6.4	13.2	0.0	6.03	4.72
		7	667	0	0.09	14.7	459.4	34.5	212.4	0.2	6.77	34.38
		9	552	0	0.13	15.8	1565	21.1	979.8	0.0	10.81	40.16
			Averages:	0	0.06	16.80	1146.95	18.45	107.70	0.25	184.58	13.10

Table 6: Comparing DMSTP methods. Instances from [33]

[7] G. Dahl, N. Foldnes, and L. Gouveia. A note on hop-constrained walk polytopes. Technical Report CIO 1, University of Lisbon, 2003.

[8] G. Dahl, L. Gouveia, and C. Requejo. On formulations and methods for the hop-constrained minimum spanning tree problem. In P. Pardalos and M. Resende, editors, *Handbook of Optimization in Telecommunications*, pages 493–515. Springer, 2006.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY,

D	$\|V\|$	OPT	LP	T(s)	Nodes	$\|V\|$	OPT	LP	T(s)	Nodes
4		747	747	0.27	1		1083	1081.5	1.32	5
5		673	673	0.38	1		977	977	3.98	1
6		606	606	4.49	1		856	856	6.86	1
7		575	575	2.14	1		821	821	5.96	1
8	41	544	544	6.05	1	61	781	781	386	1
9		532	532	2.22	3		760	760	9.87	1
10		516	516	9.73	1		734	734	91.8	1
11		508	508	0.72	1		722	722	5.91	1
12		498	498	6.55	1		712	712	7.38	1
4		1303	1303	1.39	1		1549	1549	9.58	1
5		1203	1203	8.36	1		1438	1438	362	1
6		1064	1064	226	1		1259	1259	1858	1
7		1024	1024	169	1		1220	1220	1333	1
8	81	976	976	2933	1	101	1158	1158	4574	1
9		952	952	1869	1		1136	1136	522	1
10		920	920	2341	1		1104	1104	31718	1
11		906	906	2510	1		1088	1088	42182	1
12		884	884	776	1		1060	1060	7416	1
4		1431	1431	6.82	1		1731	1731	19.7	1
5		1281	1281	42.7	1		1572	1572	8377	1
6		1059	1059	1183	1		1247	1245.83	13611	11
7		1005	1005	322	1		1177	1177	15560	1
8	121	925	925	4023	1	161	-	-	-	-
9		894	894	5636	1		-	-	-	-
10		851	851	13901	1		-	-	-	-
11		836	836	36925	1		-	-	-	-
12		812	812	29596	1		-	-	-	-

Table 7: Hop-Cut DMSTP model results over TC instances.

USA, 1979.

[10] M.T. Godinho, L. Gouveia, T. Magnanti, P. Pesneau, and J. Pires. On a time-dependent model for the unit demand vehicle routing problem. Technical Report CIO 11, University of Lisbon, 2007.

[11] L. Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers & Operations Research*, 22(9):959–970, 1995.

[12] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95(1):178–190, 1996.

[13] L. Gouveia. Using hop-indexed models for constrained spanning and steiner tree problems. In B. Samsò and P. Soriano, editors, *Telecommunications Network Planning*, pages 21–32. Springer, 1998.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

D	$\|V\|$	OPT	LP	T(s)	Nodes	$\|V\|$	OPT	LP	T(s)	Nodes
4		742	742	0.31	1		1657	1657	0.68	1
5		678	678	0.76	1		1528	1528	2.72	1
6		606	606	0.43	1		1317	1317	25.1	2
7		585	585	1.19	1		1255	1255	15.9	1
8	41	562	562	2.54	1	61	1174	1174	6.16	1
9		553	552.5	53.1	1		1139	1139	44.1	1
10		537	537	0.15	1		1083	1083	1.04	1
11		529	529	1.52	1		1063	1063	2.89	1
12		525	525	30.7	1		1044	1044	21.1	1
4		2045	2045	2.11	1		2377	2377	4.07	1
5		1828	1828	6.39	1		2166	2166	217	1
6		1564	1564	12.8	1		1802	1802	257	1
7		1479	1479	39.8	1		1708	1708	352	1
8	81	1399	1399	963	1	101	1585	1585	1037	2
9		1355	1355	3786	1		1545	1545	428	1
10		1293	1293	20.4	1		1486	1486	879	1
11		1267	1267	30.1	1		1453	1453	1302	1
12		1232	1232	309	1		1415	1415	5540	1
4		1448	1448	6.72	1		1741	1741	23.8	1
5		1297	1297	4.31	1		1583	1583	6918	1
6		1077	1077	2120	1		1253	1251.83	14422	11
7		1019	1019	333	1		1182	1182	20812	1
8	121	938	938	4213	1	161	1081	1081	158572	1
9		907	907	415	1		-	-	-	-
10		866	866	9565	1		-	-	-	-
11		848	848	4859	1		-	-	-	-
12		826	826	74178	2		-	-	-	-

Table 8: Hop-Cut DMSTP model results over TE instances.

[14] L. Gouveia. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. *INFORMS Journal on Computing*, 10(2):180–188, 1998.

[15] L. Gouveia, T. Magnanti, and C. Requejo. An intersecting tree model for odd-diameter-constrained minimum spanning and steiner trees. *Annals of Operations Research*, 146:19–39, 2006.

[16] L. Gouveia and T. L. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks*, 41(3):159–173, 2003.

[17] L. Gouveia, T. L. Magnanti, and C. Requejo. A 2-path approach for odd-diameter-constrained minimum spanning and Steiner trees. *Networks*, 44(4):254–265, 2004.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

D	$\|V\|$	OPT	LP	T(s)	Nodes	$\|V\|$	OPT	LP	T(s)	Nodes
4		249	249	0.05	1		264	264	0.15	1
5		198	198	0.43	1		213	211.25	3.34	5
6		155	155	0.24	1		155	155	7.47	1
7		144	144	0.75	1		140	140	25.1	1
8	41	135	135	0.9	1	61	124	124	7.12	1
9		131	131	0.48	1		117	117	8.75	1
10		129	129	1.04	1		114	114	8.32	1
11		128	128	2.08	1		112	111.5	13.7	1
12		127	127	0.22	1		108	108	0.57	1
4		416	416	0.93	1					
5		326	324	884	15					
6		208	208	85.8	1					
7		187	187	91.3	1					
8	81	168	167.25	8.41	1					
9		160	160	14.7	1					
10		153	153	2.39	1					
11		152	152	2.41	1					
12		151	151	1.65	1					

Table 9: Hop-Cut DMSTP model results over TR instances.

[18] L. Gouveia and P. Martins. The capacitated minimum spanning tree problem: revisiting hop-indexed formulations. *Computers & Operations Research*, 32(9):2435–2452, 2005.

[19] L. Gouveia and C. Requejo. A new lagrangian relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research*, 132(3):539–552, 2001.

[20] M. Gruber and G. R. Raidl. A new 0-1 ILP approach for the bounded diameter minimum spanning tree problem. In *Proceedings of the 2nd International Network Optimization Conference*, pages 178–185, Lisbon, 2005.

[21] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problems.* Annals of Discrete Mathematics. North-Holland, 1992.

[22] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1), 2005.

[23] B. N. Khoury, P. M. Pardalos, and D.-Z. Du. A test problem generator for the steiner problem in graphs. *ACM Trans. Math. Softw.*, 19(4):509–522, 1993.

[24] T. Koch and A. Martin. Solving steiner tree problems in graphs to optimality. *Networks*, 33:207–232, 1998.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

[25] S. Kuppa, R. Wong, and P. Thyagarajan. Load consolidation optimization of flows to a single destination. presented at 2005 INFORMS Annual Meeting, New Orleans/San Francisco, 2005.

[26] N. Maculan. The Steiner problem in graphs. *Annals of Discrete Mathematics*, 31:185–212, 1987.

[27] T. L. Magnanti and L. A. Wolsey. Optimal trees. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models*, volume 7 of *Handbook in Operations Research and Management Science*, pages 503–615. Elsevier, Amsterdam, North-Holland, 1995.

[28] P. Manyem and M. F. M. Stallmann. Some approximation results in multicasting. Technical Report TR-96-03, North Carolina State University, 1996.

[29] T. F. Noronha, A. C.Santos, and C. C. Ribeiro. Constraint programming for the diameter constrained minimum spanning tree problem. In *Proceedings of IV Latin-American Algorithms, Graphs and Optimization Symposium*, Electronic Notes in Discrete Mathematics, Puerto Varas, Chile, 2008. To appear.

[30] J. C. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110, 1978.

[31] M. Poggi de Aragão, E. Uchoa, and R. Werneck. Dual heuristics on the exact solution of large Steiner problems. *Electronic Notes in Discrete Mathematics*, 7:150–153, 2001.

[32] C.C. Ribeiro, E. Uchoa, and R.F. Werneck. A hybrid grasp with perturbations for the steiner problem in graphs. *INFORMS Journal on Computing*, 14(228–246), 2002.

[33] A. C. Santos, A. Lucena, and C. C. Ribeiro. Solving diameter constrained minimum spanning tree problems in dense graphs. In *Experimental and Efficient Algorithms*, volume 3059/2004 of *Lecture Notes in Computer Science*, pages 458–467. Springer, Berlin, 2004.

[34] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.

[35] E. Uchoa. *Algoritmos para Problemas de Steiner com Aplicações em Projeto de Circuitos VLSI.* PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2001.

[36] 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):443–472, 2008.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08

[37] R. Werneck. Problema de steiner em grafos: Algoritmos primais, duais e exatos. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2001.

[38] R. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28(3):271–287, 1984.

[39] K. A. Woolston and S. L. Albin. The design of centralized networks with reliability and availability constraints. *Computers & Operations Research*, 15(3):207–217, 1988.

Versão Final Recebida em 10/06/08 - Publicado em 24/09/08