ELSEVIER

European Journal of Operational Research 132 (2001) 539–552

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem

Luís Gouveia [a,*], Cristina Requejo [b]

[a] *DEIO/CIO, Faculdade de Ciências da Universidade de Lisboa, Bloco C2, Campo Grande, 1749-016 Lisboa, Portugal*
[b] *Departamento de Matemática/Instituto de Telecomunicações, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 3000 Coimbra, Portugal*

Received 19 July 1999; accepted 10 May 2000

Abstract

We present a new Lagrangean relaxation for the hop-constrained minimum spanning tree problem (HMST). The HMST is $\mathcal{NP}$-hard and models the design of centralized telecommunication networks with quality of service constraints. The linear programming (LP) relaxation of a hop-indexed flow-based model recently presented in the literature (see Gouveia, L., 1998. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. INFORMS Journal on Computing 10, 180–188) produces very tight bounds but has the disadvantage of being very time consuming, especially for dense graphs. In this paper, we present a new Lagrangean relaxation which is derived from the hop-indexed flow based formulation. Our computational results indicate that the lower bounds given by the new relaxation dominate the lower bounds given by previous Lagrangean relaxations. Our results also show that for dense graphs the new Lagrangean relaxation proves to be a reasonable alternative to solving the LP relaxation of the hop-indexed model.

Keywords: Design of centralized networks; Quality of service constraints; Hop constraints; Lagrangean relaxation; Multicommodity flows

1. Introduction

The hop-constrained minimum spanning tree problem (HMST) is defined as follows: Consider a graph $\mathcal{G} = (\mathcal{N}, E)$ with $\mathcal{N} = \{0, 1, \ldots, n\}$, costs c_e for each edge $e \in E$ and a natural number H. We want to find the minimum spanning tree T such that the number of hops (arcs) in the unique path from the root node, node 0, to any other node is not greater than H.

* Corresponding author. Tel.: +351-217500409; fax: +351-217500081.
E-mail address: lgouveia@fc.ul.pt (L. Gouveia).

0377-2217/01/$ - see front matter
PII: S0377-2217(00)00143-0

The HMST is $\mathcal{NP}$-hard because it contains as a particular case (the case with $H = 2$) a $\mathcal{NP}$-hard version of the simple uncapacitated facility location problem where the potential facility sites coincide with the locations of the clients to be served (see Dahl, 1998). Manyem and Stallmann (1996) have shown that the HMST is not in *APX*, i.e., the class of problems for which it is possible to have polynomial time heuristics with a guaranteed approximation bound. Dahl (1998) has studied the case with $H = 2$ from a polyhedral point of view and also compares related directed and undirected models. Gouveia (1995, 1996, 1998) has presented several lower bounding methods based on different linear integer programming models. Finally, several variations of a tabu search algorithm for obtaining upper bounds for the HMST are discussed in Voß (1999).

The HMST models the design of centralized telecommunication networks with quality of service constraints. The root node corresponds to a computer center and the remaining nodes correspond to terminals, which are required to be linked to the computer center. Hop constraints limit the number of hops (arcs) between the root node and any other node and guarantee a certain level of service with respect to some performance constraints such as *availability* and *reliability* (see Woolston and Albin, 1988; LeBlanc et al., 1998). Availability is the probability that all the transmission lines on the path from the terminal to the computer center are working. Reliability is the probability that a session will not be interrupted by a link failure. In general these probabilities decrease with the number of links in the path which means that paths with fewer hops have a better performance with respect to availability and reliability. Woolston and Albin (1988) have presented some computational results based on heuristic solutions which show that spanning tree designs with no hop limit on the path from the computer center to the terminals behave poorly with respect to reliability and availability. They have also shown that by including hop limits it is possible to generate designs with a much better service and only with a moderate increase on the total cost.

Centralized terminal networks are usually implemented with multidrop lines for connecting the terminals with the center. In such networks node processing times dominate over queuing delays and fewer hops mean, in general, lower delays.

In order to motivate the new Lagrangean relaxation, which is proposed in Section 3, Section 2 reviews the advantages and disadvantages of the lower bounding methods discussed in the literature for the HMST. Section 4 reports on our computational experience, instances corresponding to complete graphs with up to 60 nodes are used for comparing different lower-bounding schemes.

2. Formulations for the HMST

Recent advances in combinatorial optimization refer that better formulations, i.e., more compact and/or with a stronger linear programming (LP) relaxation, for several network design problems, can be obtained by defining the problem in a directed graph (see, for instance, Magnanti and Wolsey, 1995). Thus, we shall focus our discussion on directed models. A problem defined in a non-directed graph $\mathcal{G} = (\mathcal{N}, E)$ can be easily transformed into an equivalent problem in a directed graph by substituting each edge $e = \{i, j\}$ in E for two arcs, arc (i, j) and arc (j, i), and associating to each of these new arcs the cost of the original edge. Thus, the resulting graph is directed and symmetric, i.e., $c_{ij} = c_{ji}$ for all $i, j \in \mathcal{N}$. The problem of finding a minimal spanning tree in a graph is then equivalent to the problem of finding a minimal arborescence, rooted from any given node, in the equivalent directed graph. For simplicity, we assume that node 0 is the root node. We also assume that the arcs of the arborescence are directed away from the root node. Therefore, each edge $\{0, j\} \in E$ is only replaced by a single arc, arc $(0, j)$.

Furthermore, let P_{L} be the LP relaxation of formulation P and let $\vartheta(P)$ be the optimal value of P.

A directed multicommodity flow (MCF) formulation for the HMST has been presented in Gouveia (1996). The computational results given in the same work indicate that the LP bounds given by this formulation are quite good but still need to be improved, namely for instances with H small. One way of improving these

bounds has been suggested in the same work and is based on an adequate Lagrangean relaxation whose dual Lagrangean bound is strictly better than the LP bound given by the original formulation. We refer the reader to Gouveia (1996) for details about the MCF formulation, the Lagrangean relaxation and its properties. This formulation and Lagrangean relaxation will be denoted by MCF and MCF_λ, respectively, in the remainder of the text. The dual Langrangean value associated to MCF_λ will be denoted by DL.

Computational results given in Gouveia (1996) and more recently in Gouveia (1998) have shown that in general, the subgradient optimization procedure (see Held et al., 1974) used to obtain the optimal multipliers of MCF_λ needs a large number of iterations in order to produce a bound which is close to DL. One explanation for this is that when implementing the subgradient optimization procedure one needs to choose rules for initializing and updating several parameter values. Usually, the best rules depend on the problem or instance being solved and for many cases it is difficult to "tune" the method, i.e., to find a set of rules that produces good lower bounds in a reasonable amount of time. This might explain the previously described behaviour of the scheme proposed in Gouveia (1996) and why some Lagrangean-based schemes perform better than others.

To provide an alternative method for computing the value DL, Gouveia (1998) has followed the variable redefinition technique of Martin (1987) and produced a compact formulation whose LP relaxation bound is equal to the previous Lagrangean relaxation bound. The key idea for obtaining the new formulation for the HMST is the generation of a compact and exact (in the sense that the LP relaxation always gives an integer solution) formulation for the Hop-constrained shortest path (HSP) problem which arises as a subproblem in the Lagrangean relaxation MCF_λ. Then, this exact formulation was combined with the older MCF formulation and a new formulation for the HMST with a much tighter LP bound is obtained. The new formulation for the HMST uses hop-indexed flow variables and can be seen as an hop-indexed multicommodity flow (HMCF) formulation. Details about the derivation of this formulation and the exact and compact formulation for the HSP problem can be seen in Gouveia (1998).

Before describing the new model, let $\mathscr{D} = \mathscr{N} - \{0\}$ denote the set of all the nodes in G except the root node and let $\mathscr{A}$ denote the set of arcs in the underlying directed graph specified as above, i.e., $\mathscr{A} = \{(i,j) : i \in \mathscr{N},\ j \in \mathscr{D}\}$. Let $I(j) = \{i : i \in \mathscr{N} \text{ and } (i,j) \in \mathscr{A}\}$ and $J(i) = \{j : j \in \mathscr{D} \text{ and } (i,j) \in \mathscr{A}\}$, i.e., $I(j)$ denotes the set of nodes that are tails of arcs going into node j and $J(i)$ denotes the set of nodes that are heads of the arcs going out from node i.

The new model for the HMST uses the binary directed variables X_{ij} for all $(i,j) \in \mathscr{A}$, which indicate whether arc (i,j) is in the minimal spanning tree and the binary directed flow variables Z_{ijqk} for all $(i,j) \in \mathscr{A},\ k \in \mathscr{D}$ and $q = 1,\ldots,H$, which indicate whether arc (i,j) is included exactly in position q in the only path from the root node to node k.

Formulation HMCF

$$\min \quad \sum_{(i,j)\in\mathscr{A}} c_{ij}X_{ij}$$

subject to

$$\sum_{i\in I(j)} X_{ij} = 1, \quad j \in \mathscr{D}, \tag{hmcf1}$$

$$Z_{0i1k} - \sum_{j\in J(i)} Z_{ij2k} = 0, \quad i \in J(0),\ k \in \mathscr{D}, \tag{hmcf2}$$

$$\sum_{j\in I(i)} Z_{jiqk} - \sum_{j\in J(i)} Z_{ij,q+1,k} = 0, \quad i,k \in \mathscr{D},\ q = 2,\ldots,H-1, \tag{hmcf3}$$

$$\sum_{j\in I(k)\cup\{k\}} Z_{jkHk} = 1, \quad k \in \mathscr{D}, \tag{hmcf4}$$

$$\sum_{q=1}^{H} Z_{ijqk} \leqslant X_{ij}, \quad (i,j) \in \mathscr{A}, \ k \in \mathscr{D}, \tag{hmcf5}$$

$$Z_{0j1k} \in \{0,1\}, \quad j \in J(0), \ k \in \mathscr{D}, \tag{hmcf6}$$

$$Z_{ijqk} \in \{0,1\}, \quad (i,j) \in \mathscr{A}, \ i \neq 0, \ k \in \mathscr{D}, \ q = 2,\ldots,H, \tag{hmcf7}$$

$$Z_{kkqk} \in \{0,1\}, \quad k \in \mathscr{D}, \ q = 2,\ldots,H, \tag{hmcf8}$$

$$X_{ij} \in \{0,1\}, \quad (i,j) \in \mathscr{A}. \tag{hmcf9}$$

For simplicity, variables X_{ii}, for all $i \in \mathscr{D}$, variables Z_{iiqk}, for all $k \in \mathscr{D}$, $i \in \mathscr{D} - \{k\}$ and $q = 1,\ldots,H$, and variables Z_{kiqk}, for all $k, i \in \mathscr{D}$, $i \neq k$, and $q = 2,\ldots,H$, are not considered.

For each $k \in \mathscr{D}$, each system of constraints (hmcf2), (hmcf3), (hmcf4), (hmcf6), (hmcf7) and (hmcf8) is a compact and exact formulation for the HSP problem between node 0 and node k. For each k, constraints (hmcf2) state that arc $(0,i)$ in position 1 is in the path to node k if and only if there is one arc in position 2 leaving node i in the path to node k, constraints (hmcf3) state that one arc enters a given node $i \in \mathscr{D}$ in position q ($q = 2,\ldots,H-1$) if and only if one arc leaves node i in position $q+1$ and constraints (hmcf4) state that one and only one arc in position H enters node k. Constraints (hmcf5) are the linking constraints between variables X_{ij} and variables Z_{ijqk} and state that, for each $k \in \mathscr{D}$, each arc $(i,j) \in \mathscr{A}$ is included at most in one position of the HSP between node 0 and node k if the arc (i,j) is included in the solution. Notice that, for all $k \in \mathscr{D}$, each one of the HSP system uses variables Z_{kkqk} for all $q = 2,\ldots,H$. It is precisely because of the inclusion of such variables that each one of these systems allows paths between node 0 and node k with fewer than H hops. When a path contains less than H hops, self-loops are needed for obtaining a solution containing exactly H variables with value equal to 1.

To obtain the HMCF linear programming relaxation $\mathrm{HMCF_L}$, we replace constraints (hmcf6), (hmcf7), (hmcf8) and (hmcf9) with the corresponding upper and lower bounding constraints.

Gouveia (1998) has shown that

$$\mathrm{DL} = \vartheta(\mathrm{HMCF_L}),$$

i.e., the LP optimal value given by the new model is equal to the theoretical best bound associated to the Lagrangean relaxation MCF_λ.

For many cases, the LP-based approach is definitely better than the equivalent Lagrangean approach. As an example, consider the instance with 81 nodes, 280 edges and $H = 4$ reported in Gouveia (1998). The LP relaxation of the new model, $\mathrm{HMCF_L}$, used 56.9 seconds to obtain an optimal LP value of 1660 (which was also shown to be equal to the optimal integer value) while the Lagrangean relaxation MCF_λ used 5149 iterations (1438.7 seconds) on the same machine to obtain a value which is strictly greater than 1659 (thus providing the same lower bound when rounding was performed). The disadvantage of using the LP relaxation $\mathrm{HMCF_L}$ is that one might get in trouble if more dense instances or instances with a bigger number of nodes are solved. In that case, the size of the corresponding LP model might lead to huge computer storage requirements or to excessive CPU times. For such instances, the original Lagrangean relaxation MCF_λ might still be used. However it is precisely for the dense instances where MCF_λ performs poorly (mainly due to the large number of multipliers).

For completeness, we note that a Lagrangean relaxation which is quite similar to MCF_λ can also be derived from the new HMCF model. We attach non-negative multipliers δ_{ijk} for all $(i,j) \in \mathscr{A}$ and $k \in \mathscr{D}$ to the coupling constraints (hmcf5) and dualize them in the usual Lagrangean way. This Lagrangean relaxation is essentially the original Lagrangean scheme MCF_λ. The relaxed problem is decomposed into a simple inspection problem involving the X_{ij} variables and into n HSP subproblems, one for each $k \in \mathscr{D}$, involving the Z_{ijqk} variables. The relaxed problem and the dual Lagrangean problem are solved in the same way as in the case of MCF_λ and the dual Lagrangean bound of this relaxation is also equal to DL.

3. A new Lagrangean relaxation

There are two main alternatives in deriving a Lagrangean relaxation in a network flow based formulation as HMCF. One of the alternatives has just been described. The coupling constraints are dualized and the network flow structure is maintained in the relaxed problem. In the second alternative, the flow conservation constraints are dualized and the relaxed problem is usually decomposed into $|\mathscr{A}|$ subproblems, one for each arc. Although it can be proved that in general, both types of relaxations provide the same dual Lagrangean bound, computational results reported in the past for several problems (see, for instance, Beasley (1984) for the Steiner tree problem, and Gendron et al. (1999) for the capacitated network design problem) state that in general, the network flow based relaxation dominates the second type of relaxation in the sense that, for a fixed number of iterations of the subgradient optimization procedure, the first alternative provides a better bound than the second one.

These results do not necessarily imply that one should ignore the second type of relaxation. For instance, Holmberg and Yuan (1996) (see also Yuan, 1997) show that the second relaxation might provide valuable information for a branch-and-bound procedure. More recently, Crainic et al. (1999) have performed an extensive set of computations comparing bundle type methods with subgradient optimization methods for computing the optimal multipliers in the context of these two types of Lagrangean relaxations. They have shown that with an adequate "tuning" of the subgradient optimization procedure, the second type of relaxation is a viable alternative to the network flow-based relaxation. Moreover, they have shown that in the context of the second type of relaxation, the lower bounds given by the subgradient method are of the same quality as the ones obtained by using the alternative bundle type methods.

We also note that a similar relaxation has also been described in Gouveia (1996). In this relaxation, as the hop constraints have also been dualized (together with the flow conservation constraints), the associated dual Lagrangean value is shown to be equal to the LP bound given by MCF and thus, weaker than the dual Lagrangean associated to MCF_λ (the value DL). However, we have noticed that the subgradient optimization procedure associated to such relaxation was very often obtaining the theoretical best bound, the optimal value of the LP relaxation of MCF. This experiment has, in a certain way, triggered the current paper where the same type of relaxation is developed in the context of the hop-indexed HMCF formulation (that is, the hop-indexed flow conservation constraints are dualized in the HMCF formulation).

This section describes such a relaxation. We show that the associated dual Lagrangean bound is equal to DL and present computational results showing that for dense graphs it might be preferable to MCF_λ (giving an example, where a Lagrangean relaxation where the coupling constraints are dualized performs worse than a Lagrangean relaxation where the flow conservation constraints are dualized).

To derive the new Lagrangean relaxation, we attach Lagrangean multipliers γ_{i1k}, for all $i \in J(0)$ and $k \in \mathscr{D}$, to the flow conservation constraints (hmcf2), Lagrangean multipliers γ_{iqk}, for all $i,k \in \mathscr{D}$ and $q = 2,\ldots,H-1$, to the flow conservation constraints (hmcf3) and Lagrangean multipliers γ_{kHk}, for all $k \in \mathscr{D}$, to the flow conservation constraints (hmcf4) and dualize them in the usual Lagrangean way. This leads to the following relaxed problem.

Lagrangean relaxation – HMCF$_\gamma$

$$\min \quad \sum_{(i,j)\in\mathscr{A}} c_{ij}X_{ij} + \sum_{(i,j)\in\mathscr{A}} \sum_{k\in\mathscr{D}} \sum_{q=1,\ldots,H} \eta_{ijqk}Z_{ijqk} + \sum_{k\in\mathscr{D}} \gamma_{kHk}$$

subject to

$$(\text{hmcf1}), (\text{hmcf5})\text{–}(\text{hmcf9}).$$

The modified costs η_{ijqk} for all $(i,j) \in \mathscr{A},\ q = 1,\ldots,H$ and $k \in \mathscr{D}$ are given by

$$\begin{aligned}
&\eta_{0j1k} = -\gamma_{j1k} \quad \text{for } j \in J(0),\ k \in \mathscr{D},\\
&\eta_{ijqk} = \gamma_{i,q-1,k} - \gamma_{jqk} \quad \text{for } (i,j) \in \mathscr{A},\ k \in \mathscr{D},\ q = 2,\ldots,H-1,\\
&\eta_{ikHk} = \gamma_{i,H-1,k} - \gamma_{kHk} \quad \text{for } i \in \mathscr{D},\ k \in \mathscr{D}.
\end{aligned}$$

For a given value of the Lagrangean multipliers γ_{jqk} (for all $j \in J(0),\ q = 1$, and $k \in \mathscr{D}$, for all $j, k \in \mathscr{D}$ and $q = 2,\ldots,H-1$; for all $k \in \mathscr{D}$ and $q = H$), the relaxed problem HMCF$_\gamma$ can be solved using the following observation. Let $\underline{X}_{ij}$ be the optimal values of the X_{ij} $((i,j) \in \mathscr{A})$ variables in HMCF$_\gamma$. The optimal values, $\underline{Z}_{ijqk}$, of the Z_{ijqk} $(i \in \mathscr{D},\ j,k \in \mathscr{D} - \{i\}$ and $q = 1,\ldots,H)$ variables may be obtained in the following way. For each triple $\{i,j,k\}$ for all $i \in \mathscr{D}$ and $j,k \in \mathscr{D} - \{i\}$ note that by (hmcf5), (hmcf6), (hmcf7), (hmcf8) and (hmcf9) at most one of the variables Z_{ijqk}, for $q = 1,\ldots,H$, can be equal to 1. Then, let $\phi^p_{ijk} = \min_{2,\ldots,H}\{\eta_{ijqk}\}$ (i.e., p is the index for which the minimum value of this simple minimization problem occurs). If $\phi^p_{ijk} \geqslant 0$, then $\underline{Z}_{ijqk} = 0$ for all $q = 2,\ldots,H$. Otherwise, if $\phi^p_{ijk} < 0$, then $\underline{Z}_{ijpk} = \underline{X}_{ij}$ and $\underline{Z}_{ijqk} = 0$ for all $q = 2,\ldots,H,\ q \neq p$. This simply means that

$$\sum_{q=2}^{H} \eta_{ijqk}\underline{Z}_{ijqk} = \min\{0, \phi^p_{ijk}\}\underline{X}_{ij}. \tag{1}$$

In a similar way, we obtain

$$\eta_{0j1k}\underline{Z}_{0j1k} = \min\{0, \phi^p_{0jk}\}\underline{X}_{0j} \tag{2}$$

with $\phi^p_{0jk} = \eta_{0j1k}$. Thus, constraints (hmcf5), (hmcf6) and (hmcf7) can be dropped from formulation HMCF$_\gamma$ and the relaxed problem can be rewritten in the following way:

$$\min \quad \sum_{(i,j)\in\mathscr{A}} \left(c_{ij} + \sum_{k\in\mathscr{D}} \min\{0, \phi^p_{ijk}\} \right) X_{ij} + \sum_{k\in\mathscr{D}} \sum_{q=2,\ldots,H} \eta_{kkqk}Z_{kkqk} + \sum_{k\in\mathscr{D}} \gamma_{kHk}$$

subject to

$$(\text{hmcf1}),\ (\text{hmcf8}) \text{ and } (\text{hmcf9}).$$

This modified relaxation can be separated into two simple inspection subproblems. One involving only the X_{ij} $((i,j) \in \mathscr{A})$ variables and another involving only the Z_{kkqk} $(k \in \mathscr{D},\ q = 2,\ldots,H)$ variables. After the optimal $\underline{X}_{ij}$ values have been obtained, we use (1) and (2) to obtain the optimal values, $\underline{Z}_{ijqk}$, of the remaining Z_{ijqk} variables. An approximation of the optimal multipliers can be obtained by using the subgradient optimization method from Held et al. (1974).

Notice that the relaxed problem HMCF_γ satisfies the integrality property. Therefore, we have

$$\max_{\gamma \geqslant 0} \vartheta(\mathrm{HMCF}_\gamma) = \vartheta(\mathrm{HMCF}_L) = \vartheta(\mathrm{DL}),$$

and the theoretical best bound associated with this relaxation is the same as the one associated with the relaxation MCF_λ.

4. Computational results

In order to assess the performance of the new Lagrangean relaxation HMCF_γ we used the 20 and 40 node data set from Gouveia (1996) and generated new instances with 60 nodes. For each value of n, the data set contains three groups of instances. Two groups of instances are Euclidean instances and the third group refers to random instances. To obtain the Euclidean instances, the coordinates of n points corresponding to the nodes were generated accordingly to a uniform distribution on a grid with dimension 100×100 and the costs c_{ij} for each edge $\{i, j\}$ ($i = 0, 1, \ldots, n$ and $j = 1, \ldots, n$) are taken as the integer part of the Euclidean distance between points i and j generated on the grid. Two locations for the root are considered, one with the root located in the center of the grid, instances TC, and the other with the root located on a corner of the grid, instances TE. To obtain the random instances, instances TR, the costs c_{ij} for each edge $\{i, j\}$ ($i = 0, 1, \ldots, n$ and $j = 1, \ldots, n$) are randomly generated accordingly to a uniform distribution in the interval $[0, 100]$. For each value of n, and each group, TC, TE and TR, five complete graphs were considered. For each one of these instances, the hop parameter H was set to 3, 4 and 5.

The number of arcs (notice that we are using directed models) in each instance for each value of n is equal to $n + n \times (n - 1)$, hence, for each value of $n = 20, 40$ and 60, we have, respectively, 400, 1600 and 3600 arcs. In order to reduce the size of each instance, we used the following simple arc elimination test (see Gouveia (1996)). If $c_{ij} > c_{0j}$, then any optimal solution does not use 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 $c_{ij} \geqslant c_{0j}$. This arc elimination test is applied to every instance before solving each Lagrangean relaxation or solving the LP relaxation HMCF_L. Table 1 shows for each value of n and for each instance the number of arcs remaining in each instance after the elimination test was performed. Note that the test is much more effective when applied

Table 1
Size of reduced instances

n	20	40	60
TC1	153	532	1099
TC2	145	506	1164
TC3	156	533	1111
TC4	127	520	1101
TC5	127	515	1125
TE1	328	1196	2677
TE2	294	1153	2778
TE3	293	1231	2773
TE4	310	1168	2850
TE5	276	1196	2650
TR1	209	811	2083
TR2	206	819	1936
TR3	198	995	2075
TR4	228	805	1857
TR5	229	679	1853

to instances TC rather than to instances TE or TR and more effective to instances TR rather than to instances TE. This means that the reduced instances TE are larger than the reduced instances TC or TR suggesting that the TE instances will be much more difficult to solve than the remaining instances.

Each Lagrangean relaxation MCF_λ and $HMCF_\gamma$ together with the corresponding subgradient optimization method were implemented using the programming language FORTRAN. As pointed out before, for each Lagrangean relaxation and for a fixed value of the corresponding multipliers, an inspection problem on variables X_{ij} has to be solved in order to obtain the corresponding Lagrangean optimal value. Consider the well-known cut constraints

$$\sum_{i\in\mathscr{S}^c}\sum_{j\in\mathscr{S}} X_{ij} \geqslant 1 \quad \forall \mathscr{S} \subseteq \mathscr{D},\ |\mathscr{S}| \geqslant 2, \tag{3}$$

where $\mathscr{S}^c = \mathscr{N} \setminus \mathscr{S}$.

The subproblems defined in the variables X_{ij} arising in each Lagrangean relaxation become a minimal arborescence problem if such constraints are added to them. As constraints (3) are satisfied by the LP relaxation of the MCF formulation (see, for instance, Magnanti and Wolsey, 1995), the best theoretical limit obtained by solving the minimal arborescence problem is not better than the best theoretical limit obtained by solving the inspection subproblem in the variables X_{ij}. Nevertheless, by using this modified subproblem it is possible to accelerate the convergence rate of the subgradient optimization method associated to the two Lagrangean relaxations. When the multipliers are initialized to zero, the lower bound obtained at the first iteration corresponds to the value of the minimal arborescence which is, in general, much better than the optimal value of the corresponding inspection subproblem. The minimal arborescence problem can be efficiently solved using the algorithm described by Fischetti and Toth (1993). Furthermore, solving a minimal arborescence problem instead of solving an inspection subproblem allows us to use, in each iteration of the subgradient optimization method, the heuristic described in Gouveia (1996) to obtain an upper bound to the optimal value of the problem.

Different rules for initializing and updating the Lagrangean multipliers can be used for implementing the iterative subgradient optimization method proposed in Held et al. (1974). We have already referred that, usually, the best rules depend on the problem or instance being solved and for many cases it is difficult to find a set of rules that produces good lower bounds in a reasonable amount of time. Usually, the scalar used in the definition of the step size for updating the multipliers is recommended to be between 0 and 2 (Held et al., 1974). In our procedures, we noticed that the rate of convergence could be significantly improved if higher values were tried. This scalar was halved whenever the lower bound was not improved after 15 iterations of the subgradient optimization method. The maximum number of iterations for the subgradient optimization methods were fixed to 2000 for the instances with $n = 20$ and $n = 40$ and were fixed to 5000 for the instances with $n = 60$.

The computational results were obtained in an *Alpha Server* 5/300 with 128 Mb of *RAM* and are presented in Tables 2–5. The first column, denoted by Prob, identifies the problem and the second column gives the value of H. In the third column, denoted by UB, we give the value of the best upper bound value which has been obtained by the heuristic procedure included in the iterative subgradient optimization method. In some of the cases, this upper bound value is optimal (OPT). The next four columns refer to the results obtained by the Lagrangean relaxation MCF_λ. The first of these four columns, denoted by *value*, gives the best lower bound (rounded up to the next integer when not integer). The second column gives the number of iterations used by the associated subgradient optimization procedure to obtain the lower bound reported in the previous column. The third column gives the CPU time (in seconds) needed to obtain the best lower bound and the last of these four columns gives the value of the corresponding gap defined by

$$\text{gap} = \frac{UB - value}{UB} \times 100.$$

Table 2
Computational results for the instances with $n = 20$

Prob	H	OPT	MCF_λ				$HMCF_\gamma$				$HMCF_L$		
			Value	Iter	Time	Gap	Value	Iter	Time	Gap	Value	Time	Gap
TC1	3	340	338	1798	6.8	0.6	339	849	5.8	0.3	339	6.33	0.3
	4	318	318	1651	7.5	0.0	318	799	9.0	0.0	318	53.78	0.0
	5	312	312	1498	8.0	0.0	312	801	13.8	0.0	312	94.78	0.0
TC2	3	365	362	1700	6.4	0.8	364	697	4.8	0.3	364	8.75	0.3
	4	338	338	648	2.9	0.0	338	293	3.6	0.0	338	143.35	0.0
	5	332	332	1038	5.5	0.0	332	550	9.3	0.0	332	64.30	0.0
TC3	3	343	337	1800	6.9	1.7	339	797	5.6	1.2	339	2.47	1.2
	4	306	305	1598	7.3	0.3	306	701	8.8	0.0	306	57.87	0.0
	5	296	296	1148	6.1	0.0	296	650	11.3	0.0	296	83.03	0.0
TC4	3	390	387	1098	4.0	0.8	387	196	1.3	0.8	387	10.50	0.8
	4	376	374	1348	6.0	0.5	376	1649	19.3	0.0	376	50.45	0.0
	5	364	364	1549	8.1	0.0	364	1100	17.6	0.0	364	102.73	0.0
TC5	3	347	346	1898	7.0	0.3	347	250	1.7	0.0	347	5.02	0.0
	4	326	323	1650	7.4	0.9	324	650	7.6	0.6	324	41.20	0.6
	5	310	310	1	0.004	0.0	310	1	0.001	0.0	310	50.67	0.0
TE1	3	449	434	1950	9.2	3.3	441	1199	7.7	1.8	444	650.55	1.1
	4	385	377	1950	10.7	2.1	382	1800	21.1	0.8	385	2743.17	0.0
	5	366	356	1701	10.6	2.7	358	1551	25.7	2.2	364	6204.72	0.5
TE2	3	435	434	1951	8.8	0.2	435	347	2.1	0.0	435	338.37	0.0
	4	404	395	1748	9.3	2.2	398	1501	16.7	1.5	401	1970.23	0.7
	5	383	374	1650	10.0	2.3	376	1699	29.5	1.8	381	4442.85	0.5
TE3	3	435	432	1948	8.8	0.7	435	585	3.6	0.0	435	294.77	0.0
	4	396	392	1899	10.1	1.0	394	1300	14.6	0.5	395	1009.27	0.3
	5	372	371	1998	12.1	0.3	371	1650	26.1	0.3	372	2634.52	0.0
TE4	3	448	438	1900	8.8	2.2	443	998	6.3	1.1	444	410.45	0.9
	4	402	391	1998	10.8	2.7	396	1949	22.3	1.5	400	1682.70	0.5
	5	382	371	1900	11.7	2.9	373	1901	30.3	2.4	377	2707.10	1.3
TE5	3	428	424	1849	8.2	0.9	427	949	5.7	0.2	428	123.62	0.0
	4	376	376	1700	8.9	0.0	376	648	7.1	0.0	376	2120.30	0.0
	5	354	353	1251	7.5	0.3	354	1301	19.8	0.0	354	4562.47	0.0
TR1	3	168	166	1932	7.8	1.2	168	247	1.4	0.0	168	8.52	0.0
	4	146	145	1782	8.6	0.7	146	297	2.9	0.0	146	18.05	0.0
	5	137	137	1	0.005	0.0	137	1	0.01	0.0	137	88.30	0.0
TR2	3	201	201	1501	6.1	0.0	201	144	0.8	0.0	201	62.13	0.0
	4	161	159	1936	9.3	1.2	161	847	8.3	0.0	161	522.88	0.0
	5	140	140	1195	6.7	0.0	140	343	4.6	0.0	140	1046.88	0.0
TR3	3	157	152	1648	6.6	3.2	153	332	1.8	2.5	153	19.58	2.5
	4	130	130	852	4.1	0.0	130	188	1.8	0.0	130	60.50	0.0
	5	121	121	1	0.005	0.0	121	1	0.01	0.0	121	336.50	0.0
TR4	3	183	181	1797	7.5	1.1	183	447	2.5	0.0	183	69.15	0.0
	4	158	158	2000	9.9	0.0	158	499	5.1	0.0	158	554.37	0.0
	5	142	142	846	4.9	0.0	142	250	3.5	0.0	142	1353.32	0.0
TR5	3	148	144	1949	8.2	2.7	148	237	1.4	0.0	148	88.60	0.0
	4	121	118	1700	8.5	2.5	120	1100	11.1	0.8	121	584.58	0.0
	5	110	107	1853	10.6	2.7	109	1598	22.4	0.9	110	2100.82	0.0

Table 3
Computational results for the instances with $n = 40$

Prob	H	UB	MCF_λ				$HMCF_\gamma$				$HMCF_L$		
			Value	Iter	Time	Gap	Value	Iter	Time	Gap	Value	Time	Gap
TC1	3	613	566	1992	84.2	7.7	604	898	58.1	1.5	605	9835.2	1.3
	4	549	518	1947	94.6	5.6	546	1593	151.5	0.5	547	63739.4	0.4
	5	522	499	1900	104.0	4.4	518	1698	226.7	0.8	522	223904.8	0.0
TC2	3	566	539	1943	81.9	4.8	565	389	25.0	0.2	566	11398.1	0.0
	4	519	498	1998	96.6	4.0	515	1994	188.7	0.8	518	119644.5	0.2
	5	496	481	1949	106.3	3.0	492	1595	210.6	0.8	494	73723.6	0.4
TC3	3	580	549	1939	82.4	5.3	579	542	35.7	0.2	580	5866.1	0.0
	4	547	512	1949	94.9	6.4	534	1947	187.1	2.4	538	73665.4	1.6
	5	522	493	1850	101.5	5.6	509	1398	187.5	2.5	514	313660.7	1.5
TC4	3	627	569	1998	84.6	9.3	608	1849	120.6	3.0	609	11203.9	2.9
	4	571	520	1998	97.0	8.9	547	1848	174.7	4.2	552	82753.1	3.3
	5	532	500	1950	106.3	7.1	520	1644	216.6	2.3	522	246727.5	1.9
TC5	3	602	562	1943	81.6	6.6	593	1643	106.9	1.5	594	11730.3	1.3
	4	567	522	1998	96.3	7.9	544	1188	111.1	4.1	548	86169.9	3.4
	5	524	501	1943	105.6	4.4	519	1386	178.5	1.0	522	351232.6	0.4
TE1	3	722	649	1997	99.8	10.1	697	1350	112.3	3.5	–	+259200.0	–
	4	654	575	1950	109.5	12.1	616	1939	238.4	5.8			
	5	610	543	1951	121.7	11.0	573	1500	248.9	6.1			
TE2	3	796	653	1949	96.4	18.0	695	1500	123.9	12.7	–	+172800.0	–
	4	627	565	2000	111.4	9.9	612	1486	178.6	2.4			
	5	609	537	1950	121.1	11.8	564	1700	279.3	7.4			
TE3	3	692	614	2000	101.0	11.3	655	1700	142.6	5.3			
	4	606	534	1948	110.5	11.9	571	1246	153.5	5.8			
	5	549	502	1950	122.7	8.6	528	1943	333.8	3.8			
TE4	3	749	660	1995	99.4	11.9	710	1591	129.9	5.2			
	4	629	578	1950	109.3	8.1	618	1799	218.9	1.7			
	5	594	548	2000	124.6	7.7	573	1643	272.7	3.5			
TE5	3	675	618	1994	100.0	8.4	674	929	77.0	0.1			
	4	643	551	1997	113.1	14.3	597	1799	220.2	7.2			
	5	591	518	1999	125.5	12.4	552	1899	313.3	6.6			
TR1	3	176	164	1986	91.1	6.8	175	523	38.1	0.6	176	10334.2	0.0
	4	149	140	1888	98.3	6.0	148	840	88.3	0.7	149	110654.4	0.0
	5	139	132	1588	92.4	5.0	138	797	115.3	0.7	–	+208800.0	–
TR2	3	219	197	1946	89.2	10.0	218	928	67.1	0.5	219	16856.38	0.0
	4	176	159	1989	103.6	9.7	175	876	91.5	0.6			
	5	155	147	1793	104.4	5.2	154	1080	155.5	0.6			
TR3	3	198	174	1951	93.6	12.1	197	477	36.9	0.5	198	44260.37	0.0
	4	145	130	1934	104.0	10.3	144	1226	137.5	0.7			
	5	123	116	1683	100.8	5.7	122	1000	154.0	0.8			
TR4	3	167	155	1887	85.4	7.2	166	184	13.4	0.6	167	10821.65	0.0
	4	131	125	1896	97.6	4.6	130	217	22.9	0.8			
	5	122	117	1451	83.6	4.1	121	1045	152.6	0.8			
TR5	3	205	185	1983	86.9	9.8	204	275	19.0	0.5	205	3226.72	0.0
	4	159	151	1935	97.3	5.0	158	343	34.5	0.6			
	5	149	141	1639	92.0	5.4	148	848	118.6	0.7			

Table 4
Computational results with a maximum of 5000 iterations for some instances with $n = 20$ and $n = 40$

n	Prob	H	UB	MCF_λ				$HMCF_\gamma$			
				Value	Iter	Time	Gap	Value	Iter	Time	Gap
20	TC1	3	340	339	3151	12.1	0.3	339	849	4.3	0.3
		4	318	318	1651	7.6	0.0	318	799	7.2	0.0
		5	312	312	1498	8.0	0.0	312	801	9.7	0.0
	TE1	3	449	439	4601	21.8	2.2	442	2099	13.5	1.6
		4	385	381	4500	24.9	1.0	384	4299	51.1	0.3
		5	366	359	4750	29.8	1.9	361	5000	82.6	1.4
	TR1	3	168	168	2481	10.1	0.0	168	247	1.4	0.0
		4	146	146	2185	10.6	0.0	146	297	2.9	0.0
		5	137	137	1	0.003	0.0	137	1	0.02	0.0
40	TC1	3	613	590	4849	205.2	3.8	604	898	58.4	1.5
		4	549	533	4750	231.8	2.9	547	3344	314.8	0.4
		5	522	509	4948	276.7	2.5	520	2938	376.9	0.4
	TE1	3	722	684	4901	247.0	5.3	698	3600	297.9	3.3
		4	654	602	4801	270.1	8.0	620	4050	495.4	5.2
		5	610	564	4900	306.0	7.5	581	5000	820.3	4.8
	TR1	3	176	172	4440	202.2	2.3	175	523	38.0	0.6
		4	149	146	4801	248.5	2.0	148	840	88.8	0.7
		5	139	136	4048	234.4	2.2	138	797	114.6	0.7

The next four columns are similar and refer to the Lagrangean relaxation $HMCF_\gamma$. The last three columns refer to the LP bounds given by HMCF (again, rounded up to the next integer when not integer), the corresponding CPU time and gap.

The reported results indicate that the lower bounds obtained by the Lagrangean relaxation $HMCF_\gamma$ are a lot better than the lower bounds obtained by the Lagrangean relaxation MCF_λ. Notice also that the number of iterations of the subgradient optimization method associated to the Lagrangean relaxation $HMCF_\gamma$ is substantially smaller than the number of iterations of the subgradient optimization method associated to the Lagrangean relaxation MCF_λ. This can be explained by the fact that only $\mathcal{O}(n^2H)$ multipliers are involved in $HMCF_\gamma$ while $\mathcal{O}(n|\mathscr{A}|)$ multipliers are involved in MCF_λ. In the case of dense graphs, the number of multipliers in the relaxation $HMCF_\gamma$ is significantly lower than the number of multipliers in the relaxation MCF_λ. This suggests that the number of iterations needed by the subgradient optimization method to obtain a reasonable lower bound may be smaller with the new Lagrangean relaxation $HMCF_\gamma$ (as we said before, this is confirmed by our computational results). However, the CPU times produced by $HMCF_\gamma$ are, in general, bigger than the CPU times produced by MCF_λ. The reason for this is that the CPU time used in each iteration of the new method is substantially bigger than the CPU time used in each iteration of the older relaxation. At first sight, this seems to be unexpected. Notice that only two simple inspection problems and a simple minimization problem (with complexity $\mathcal{O}(|\mathscr{A}|H)$ to compute (1) and (2)) have to be solved in each iteration of the subgradient optimization procedure associated with $HMCF_\gamma$, while one simple inspection problem and n hop-constrained shortest path problems have to be solved in each iteration of the subgradient optimization procedure associated with MCF_λ. This suggests that one iteration of the subgradient optimization method might be significantly faster with the new relaxation. However, as we have said before, this is not confirmed by our computational results. Our explanation for this is that variables with four indexes are involved in $HMCF_\gamma$ and the computer programs implemented for solving the subproblems associated with $HMCF_\gamma$ involve four nested cycles (notice that only three nested cycles are needed in the other relaxation).

Table 5
Computational results for the instances with $n = 60$

Prob	H	UB	MCF_λ				$HMCF_\gamma$			
			Value	Iter	Time	Gap	Value	Iter	Time	Gap
TC1	3	894	823	4999	888.0	7.9	861	2950	772.9	3.7
	4	805	740	4899	997.0	8.1	775	4199	1842.7	3.7
	5	734	705	4900	1089.9	4.0	731	4839	3013.7	0.4
TC2	3	845	795	4949	876.8	5.9	844	1299	347.1	0.1
	4	767	718	4901	981.3	6.4	763	4521	2014.0	0.5
	5	756	691	4849	1075.8	8.6	723	4949	3115.0	4.4
TC3	3	915	837	4949	878.8	8.5	880	2847	751.4	3.8
	4	826	761	4948	982.7	7.9	798	4237	1859.8	3.4
	5	789	729	4850	1063.7	7.6	759	3250	2433.1	3.8
TC4	3	880	807	4949	875.9	8.3	854	3436	901.3	3.0
	4	826	733	4901	968.7	11.3	770	2840	1300.2	6.8
	5	747	693	4751	1026.5	7.2	722	4694	3513.4	3.3
TC5	3	1021	937	4998	868.4	8.2	990	3139	803.2	3.0
	4	964	845	4948	966.3	12.3	890	3848	1764.5	7.7
	5	879	799	4851	1051.5	9.1	841	4339	3232.3	4.3
TE1	3	1707	1409	4998	1079.3	17.5	1497	4499	1470.1	12.3
	4	1503	1202	4999	1190.1	20.0	1296	3899	2108.5	13.8
	5	1341	1115	5000	1294.4	16.9	1188	4000	2976.6	11.4
TE2	3	1738	1376	4950	1082.2	20.8	1436	3900	1277.3	17.4
	4	1432	1163	4951	1188.3	18.8	1250	4645	2540.6	12.7
	5	1317	1084	4998	1305.6	17.7	1145	4447	3340.3	13.1
TE3	3	1572	1315	4998	1088.2	16.3	1419	4848	1605.6	9.7
	4	1426	1123	4950	1185.2	21.2	1209	4598	2525.6	15.2
	5	1259	1038	4900	1282.2	17.6	1112	4449	3363.6	11.7
TE4	3	1811	1399	5000	1101.7	22.7	1480	4647	1538.8	18.3
	4	1454	1173	4901	1186.8	19.3	1262	4846	2672.6	13.2
	5	1320	1092	4901	1286.7	17.3	1152	4999	3791.2	12.7
TE5	3	1791	1385	4950	1063.4	22.7	1459	4649	1501.3	18.5
	4	1499	1166	4900	1156.5	22.2	1256	3694	2004.3	16.2
	5	1328	1081	4998	1286.4	18.6	1155	4396	3256.7	13.0
TR1	3	274	247	4837	970.3	9.9	273	535	162.0	0.4
	4	207	194	4549	1002.1	6.3	206	594	296.5	0.5
	5	190	183	4900	1185.3	3.7	188	1748	1216.4	1.1
TR2	3	243	216	4844	945.3	11.1	242	831	247.5	0.4
	4	186	168	4983	1080.2	9.7	183	1848	904.1	1.6
	5	165	154	4487	1065.9	6.7	164	1343	921.4	0.6
TR3	3	288	261	4835	961.2	9.4	287	490	150.0	0.3
	4	235	209	4748	1045.4	11.1	225	2199	1156.1	4.3
	5	227	193	4401	1066.3	15.0	204	4600	3308.9	10.1
TR4	3	204	185	4783	926.0	9.3	203	385	113.6	0.5
	4	163	155	4700	1009.9	4.9	162	488	237.1	0.6
	5	151	148	4851	1144.5	2.0	150	545	371.3	0.7
TR5	3	230	212	4985	962.6	7.8	229	283	83.5	0.4
	4	180	173	4748	1017.4	3.9	179	349	169.8	0.6
	5	172	162	4048	954.9	5.8	169	2291	1652.4	1.7

As expected, the lower bounds obtained by $HMCF_L$ cannot be worse than the lower bounds obtained by $HMCF_\gamma$. However, the CPU times produced by the $HMCF_L$ relaxation are, in general, bigger than the CPU times produced by the Lagrangean relaxation $HMCF_\gamma$.

For several instances with $n = 20$ (see Table 2), the optimal value was found (i.e. the reported lower bound equals the optimal value obtained by a branch and bound procedure using the *CPLEX* routine). For all instances TC and most instances TR, the lower bounds obtained by $HMCF_L$ are equal to the lower bounds obtained by $HMCF_\gamma$. For the instances TE, the lower bounds obtained by $HMCF_\gamma$ are better than the lower bounds obtained by MCF_λ. Additionally, the lower bounds obtained by $HMCF_\gamma$ are not as good as the lower bounds obtained by $HMCF_L$. However, the CPU times of the LP relaxation are bigger than the CPU times produced by both Lagrangean relaxations.

For instances with $n = 40$ (see Table 3), the lower bounds obtained by $HMCF_\gamma$ are better than the lower bounds obtained by MCF_λ and the subgradient optimization method associated to relaxation $HMCF_\gamma$ uses fewer iterations than the subgradient optimization method associated to relaxation MCF_λ. Again, the lower bounds obtained by $HMCF_L$ are better than the lower bounds obtained by both Lagrangean relaxations after the limit of 2000 iterations. However, for most of the cases the LP relaxation takes huge CPU times to obtain those values (the LP optimal value was not obtained after a couple of days of computations for most of the instances).

In all the instances tested, the CPU times produced by $HMCF_\gamma$ are bigger than the CPU times produced by MCF_λ. Thus, we have increased the number of iterations to 5000 in MCF_λ in order to compare the bounds produced by MCF_λ after 5000 iterations with the bounds produced by $HMCF_\gamma$ after 2000 iterations. These results refer only to some of the 20 and 40 node instances and are presented in Table 4. The results show that the new relaxation still produces better lower bounds than the older relaxation. Reasonable improvements with respect to the new relaxation $HMCF_\gamma$ are also obtained when the number of iterations was also increased to 5000. See, again Table 4.

With respect to the instances with $n = 60$ (see Table 5), it is interesting to point out that for most of the instances TE with $H = 3$ the CPU time used by the Lagrangean relaxation $HMCF_\gamma$ is smaller than the CPU time used by the older Lagrangean relaxation. A similar situation happens in some of the instances TR. Again, notice that the lower bounds produced by the new Lagrangean relaxation are in general much better than the lower bounds given by the older Lagrangean relaxation. The LP relaxation was not tested for the $n = 60$ instances. The reason for this was the huge CPU times obtained for the $n = 40$ instances.

5. Conclusions

In this paper, we have presented a new Lagrangean relaxation for the hop-constrained minimum spanning tree problem. Our computational experience shows that for instances with a large number of edges, the new Lagrangean relaxation is superior to a different Lagrangean relaxation previously presented in the literature. The results also show that for the same class of instances the new Lagrangean relaxation is a sound alternative to solving directly the LP relaxation of the same model.

Acknowledgements

The authors thanks the referees for their suggestions which have helped to improve the paper. Thanks also to Bernard Gendron for pointing out his latest work with Teodor Crainic and Antonio Frangioni.

References

Beasley, J., 1984. An algorithm for the Steiner problem in graphs. Networks 14, 147–159.

Crainic, T.G., Frangioni, A., Gendron, B., 1999. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. Discrete Applied Mathematics 112, 73–99.

Dahl, G., 1998. The 2-hop spanning tree problem. Operations Research Letters 23, 21–26.

Fischetti, M., Toth, P., 1993. An efficient algorithm for the min-sum arborescence problem on complete digraphs. ORSA Journal on Computing 5, 426–434.

Gendron, B., Crainic, T.G., Frangioni, A., 1999. Multicommodity capacitated network design. In: Sansò, B., Soriano, P. (Eds.), Telecommunications Network Planning. Kluwer Academic Publishers, Dordrecht.

Gouveia, L., 1995. Using the Miller–Tucker–Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. Computer and Operations Research 22, 959–970.

Gouveia, L., 1996. Multicommodity flow models for spanning trees with hop constraints. European Journal of Operational Research 95, 178–190.

Gouveia, L., 1998. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. INFORMS Journal on Computing 10, 180–188.

Held, M., Wolfe, P., Crowder, H.P., 1974. Validation of subgradient optimization. Mathematical Programming 6, 62–88.

Holmberg, K., Yuan, D., 1996. A Lagrangean heuristic based branch-and-bound approach for the capacitated network design problem. Research Report LiTH-MAT-R-1996-23, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden. Operations Research (accepted).

LeBlanc, L., Chifflet, J., Mahey, P., 1998. Packet routing in telecommunication networks with path and flow restrictions. INFORMS Journal on Computing (to appear).

Magnanti, T., Wolsey, L., 1995. Optimal trees. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Network Models, Handbooks in Operations Research and Management Science, Vol. 7. North-Holland, Amsterdam.

Manyem, P., Stallmann, M., 1996. Some Approximation Results in Multicasting. Working Paper, North Carolina State University.

Martin, R.K., 1987. Generating alternative mixed-integer programming models using variable redefinition. Operations Research 35, 820–831.

Voß, S., 1999. The Steiner tree problem with hop constraints. In: Sharaiha, Y., Beasley, J. (Eds.), Advances in Combinatorial Optimization. Annals of Operations Research, vol. 86. pp. 271–294.

Woolston, K., Albin, S., 1988. The design of centralized networks with reliability and availability constraints. Computer and Operations Research 15, 207–217.

Yuan, D., 1997. Optimal Synthesis and Multicommodity Routing in Telecommunication Networks. PhD thesis, Division of Optimization, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden.