Models and Branch-and-Cut Algorithms for the Steiner Tree Problem with Revenues, Budget and Hop Constraints

Alysson M. Costa, Jean-François Cordeau, and Gilbert Laporte
HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

The Steiner tree problem with revenues, budget and hop constraints is a variant of the Steiner tree problem with two main modifications: (a) besides the costs associated with arcs, there are also revenues associated with the vertices, and (b) there are additional budget and hop constraints, which impose limits on the total cost of the network and on the number of edges between any vertex and the root, respectively. This article introduces and compares several mathematical models for this problem and describes two branch-and-cut algorithms, which solve to optimality instances with up to 500 vertices and 625 edges. © 2008 Wiley Periodicals, Inc. NETWORKS, Vol. 53(2), 141–159 2009

Keywords: prize collecting; network design; Steiner tree problem; budget; branch-and-cut; hop constraints

1. INTRODUCTION

Several real-life decision situations can be described as the problem of determining a least cost network spanning all or some of the vertices of a graph. The most widely known cases include the Steiner Tree Problem (STP) and the Minimum Spanning Tree Problem (MSTP). These two problems are described as follows. Let $G = (V, E)$ be a graph with vertex set $V = \{1, \dots, n\}$, where vertex 1 is the *root* vertex, and edge set $E = \{e = (i,j) : i,j \in V, i < j\}$, where each edge $e \in E$ has an associated cost c_e. The set V is partitioned into a set of *terminal* vertices (including the root) and a set of *Steiner* vertices. The STP consists of determining a minimum cost tree spanning all terminal vertices and possibly some Steiner vertices (see, for instance, [8, 17, 26]). The MSTP is a special case of the STP for which all vertices are terminal. Unlike the MSTP, which can be solved in polynomial time (see, e.g., [37]), the STP is NP-hard [16].

Received May 2006; accepted July 2007
Correspondence to: G. Laporte; e-mail: gilbert@crt.umontreal.ca
Contract grant sponsor: Brazilian Federal Agency for Post-Graduate Education-CAPES; Contract grant number: BEX 1097/01-06
Contract grant sponsor: Canadian Natural Science and Engineering Research Council; Contract grant numbers: 227837-04 and 39682-05
DOI 10.1002/net.20274
Published online 17 December 2008 in Wiley InterScience (www.interscience.wiley.com).

Steiner Tree Problems with Revenues (STPR) are an important generalization of the classical STP. In the STPR, in addition to the costs associated with the edges, there is also a revenue $r_i \geq 0$ associated with each vertex i. The goal is to determine a cost minimizing or revenue maximizing subtree subject to constraints. This article deals with a particular case of the STPR with two main features. We are first interested in the *Steiner Tree Problem with Revenues and Budget* (STPRB), where the goal is to maximize the collected revenue whereas respecting an upper limit on the total network cost. We also consider hop constraints, which limit the number of edges between any vertex in the solution and the root vertex to an upper limit equal to h. We call the resulting problem the *Steiner Tree Problem with Revenues, Budget and Hop Constraints* (STPRBH).

Several authors (see, e.g., [31, 33]) have studied a related version of the STPR where the goal is to maximize the difference between the collected revenues and the edge costs. This problem is often called the *Prize-Collecting Steiner Tree Problem* or the *Steiner Tree Problem with Profits* (STPP). Because, in our case, the goal is to maximize the collected revenue whereas respecting a limit on the total cost (and not to maximize the difference between the collected revenue and the network cost), we use the term "revenue" to avoid confusion. Maximizing revenue and considering the costs only in the constraints instead of maximizing the net profit may be useful, for example, when costs and revenues are measured in incommensurate units (e.g. resource and monetary units).

Our motivation for studying the STPRBH is twofold. Concerning the budget constraint, we have been stimulated both by the lack of studies dealing with this type of constraint and by its practical importance. Indeed, we are aware of only two articles that have considered such constraints [27, 39], despite the fact that they are often present in real-life situations [7]. Hop constraints, in turn, are often used in telecommunications applications. They guarantee that the probability of service failure at a vertex will not exceed a given threshold $1 - (1 - \pi)^h$, where π is the probability of failure of any link and h is the number of allowed hops. These constraints also restrict the maximum transmission delay in telecommunication networks [22]. Finally, Voß [40] mentions a different motivation for considering Steiner trees with hop constraints: when modeling certain lot sizing problems as Steiner tree problems, hop constraints limit the number of periods during which some goods can be held in stock.

Steiner Tree Problems with Profits have received considerable attention by operations researchers [7]. The first studies go back to Segev [38] who implemented simple greedy heuristics and a Lagrangean lower bounding procedure. Bienstock et al. [2] have proposed a heuristic with worst case performance ratio of 3, whereas Goemans and Williamson [19], Johnson et al. [27], and Cole et al. [6] have developed approximation algorithms. More recent research has focused on lower bounding procedures and metaheuristics. Canuto et al. [3] have proposed a multi-start algorithm with local search and a variable neighborhood search postoptimization phase. Cunha et al. [9] have developed a Lagrangean relaxation algorithm with dynamically generated cutting planes, whereas Klau et al. [29] have proposed a hybrid exact-memetic algorithm where the final population of the evolutionary approach is used to construct a reduced instance that will go through an exact optimization phase. Two recent articles have applied branch-and-cut algorithms to the STPP. Lucena and Resende [33] have obtained interesting results based on a separation of generalized subtour constraints. These results have later been improved by Ljubić et al. [31] who have dealt with an alternative formulation and proposed a separation procedure to identify violated connectivity constraints. An interesting variant of the STPP, called the *Prize Collecting Generalized Minimum Spanning Tree Problem*, was recently proposed by Golden et al. [20]. In this problem the vertices are partitioned into clusters and a revenue is associated with each vertex. The problem consists of determining a tree of maximal profit spanning all clusters. The authors present several heuristic strategies and a branch-and-cut algorithm that solves instances containing up to 200 vertices.

Despite the considerable amount of research dedicated to the STPP, very little work has been done on the STPRB. Indeed, we are only aware of the article of Johnson et al. [27] who proposed an approximation algorithm with limited practical interest. This lack of research is particulary intriguing due to the fact that budgetary constraints are often present in real-life situations. Indeed, Costa et al. [7] refer to several articles in which budget constraints are considered in the context of the *Traveling Salesman Problem with Profits*. As discussed in their article, the relevance of these constraints extends to the STPP. This article contains an example along the lines of Johnson et al. [27] in which the use of the budgetary constraints helps differentiate solutions having the same objective function given by the *profit minus the cost*, but with very different practical implications.

Concerning the hop constraints no author has, to our knowledge, ever considered their inclusion (or the inclusion of any other notion of reliability) in conjunction with the STPP. These constraints have only been considered in more classical contexts such as the MST and STP. Gouveia [22] and Gouvia and Requejo [25] have proposed Lagrangean relaxation lower bounding approaches for the MST with hop constraints and were able to solve instances on complete graphs of up to 60 vertices. Gouveia [24] has presented several hop-indexed models for the MST and STP with hop constraints, among other problems, whereas Dahl et al. [10] have described a general framework for modeling hop constrained MST problems. Finally, Voß [40] has presented a mathematical formulation and a tabu search for the STP with hop constraints.

We study four mathematical models for the STPRBH, and we propose several branch-and-cut algorithms. Our choice of branch-and-cut methods is motivated by the good results obtained by Lucena and Resende [33] and Ljubić et al. [31] in the context of the pure STPP. The first algorithm relies on the initial relaxation of connectivity and hop constraints, whereas the second relaxes a set of linking constraints on disaggregated variables. In both cases, the constraints are dynamically included in the model when found to be violated. Our computational experiments show that the hop constraints make the problem highly difficult and that the choice of the most efficient strategy depends on the maximum number of hops.

The remainder of this article is organized as follows. Section 2 proposes mathematical models for the STPRBH. In Section 3, we present the branch-and-cut algorithms. Section 4 reports the computational results and the article ends with some conclusions in Section 5.

2. FORMULATIONS

We introduce four formulations for the STPRB. In Section 2.1, we adapt the Dantzig-Fulkerson-Johnson (DFJ) subtour elimination constraints and connectivity constraints [11] to construct an undirected and a directed formulation for the STPRBH, respectively. The resulting formulations contain a limited number of variables but an exponential number of constraints. A different approach is used in Section 2.2, where we use a lifted version of the Miller-Tucker-Zemlin (MTZ) constraints [12, 36] to eliminate circuits and to limit the number of hops in each path of the solution. Finally, in Section 2.3 we model the problem with three-index position variables, and we show that this formulation dominates that obtained with the MTZ constraints.

Gouveia [22] also presents a formulation belonging to a different family and based on multicommodity flow variables for MST problems with hop constraints, which could also be adapted to the STPRBH. A variation of the multicommodity flow formulation has been proposed by Gouveia [23] and Gouvia and Requejo [25], inspired by the idea of variable redefinition. This latter formulation is a combination of an exact solution for the hop-constrained shortest path problem and the original multicommodity flow formulation. In both cases, the authors have proposed the use of Lagrangean relaxation methods to solve the formulations. Computational tests have shown that even their linear relaxations pose a real challenge. For this reason, we do not consider formulations of this type.

2.1. Dantzig-Fulkerson-Johnson Formulations

Several mathematical formulations have been developed for the STPP and can be adapted to the STPRB. The following model is based on the classical subtour elimination constraints introduced by Dantzig et al. [11] for the *Travelling Salesman Problem* (TSP). Let x_{ij} and y_i be binary variables associated with edges $(i,j) \in E$ and vertices $i \in V$, respectively. Variable y_i is equal to 1 if vertex i belongs to the solution ($y_1 = 1$), and to 0 otherwise. Likewise, variable x_{ij} is equal to 1 if edge (i,j) belongs to the solution, and to 0 otherwise. For $S \subseteq V$, define $E(S)$ as the set of edges with both end vertices in S. Let also $P = (i_1 = 1, \ldots, i_\ell)$ denote a path originating at the root node and containing ℓ vertices. Finally, define $\mathcal{P}_h$ as the set of paths P in G with length $\ell = h + 2$. Each path $P \in \mathcal{P}_h$ contains $h + 1$ edges and thus violates the hop constraint. The STPRBH can then be written as:

Undirected Dantzig-Fulkerson-Johnson (UDFJ) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{1}$$

subject to

$$\sum_{(i,j)\in E} x_{ij} = \sum_{i \in V} y_i - 1, \tag{2}$$

$$\sum_{(i,j)\in E(S)} x_{ij} \leq \sum_{i \in S \setminus \{k\}} y_i, \quad k \in S \subseteq V, |S| \geq 2, \tag{3}$$

$$\sum_{(i,j)\in E} c_{ij} x_{ij} \leq b, \tag{4}$$

$$\sum_{t=2}^{h+2} x_{i_{t-1},i_t} \leq h, \quad P = (i_1 = 1, \ldots, i_{h+2}) \in \mathcal{P}_h, \tag{5}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in E, \tag{6}$$

$$y_1 = 1, \tag{7}$$

$$y_i \in \{0,1\}, \quad i \in V \setminus \{1\}. \tag{8}$$

The objective function maximizes the revenue of the spanned vertices. Constraint (2) forces the presence of $v - 1$ edges connecting the v spanned vertices. Constraints (3) are generalized subtour elimination constraints. These are stronger than the classical subtour elimination constraints used in the TSP formulation in which the right-hand side is $|S| - 1$. Note that for subsets S formed only by spanned vertices, constraints (3) reduce to the classical subtour elimination constraints. Constraint (7) forces the presence of the root vertex in the solution. Constraints (4) and (5) are specific to the STPRBH. Constraint (4) is the budget constraint, which forces the total network cost not to exceed the budget b, whereas constraints (5) guarantee that there are no more than h hops between the root vertex and any other vertex in the solution.

A version of model (1)–(8) with no budget and no hop constraints was used by Resende [33] and follows from an extended formulation for the STP proposed by Lucena [32], Goemans [18], and Margot et al. [35]. It is interesting to observe that in the absence of hop constraints, the x variables associated with edges need not be declared as integers. Indeed, when the y variables are equal to 0 or 1, constraints (2) and (3) define the convex hull of the characteristic vectors of the spanning trees on the subgraph of G induced by the selected vertices [35].

Formulation (1)–(8) is a straightforward model for the STPRBH. An equivalent model can be developed based on constraints that ensure the existence of a path between the root vertex and all other selected vertices. This has been done in the context of the STPP by Ljubić et al. [31]. Consider an arc set A containing two directed arcs (i,j) and (j,i) for each edge $(i,j) \in E$, but no arc entering the root. A directed rooted tree is called a *Steiner arborescence*. Given a directed set of arcs, the STPRBH can be written as the problem of finding a Steiner arborescence rooted at vertex 1 and spanning a subset Y of vertices with maximum revenue $\sum_{i \in Y} r_i$.

Directed Dantzig-Fulkerson-Johnson (DDFJ) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{9}$$

subject to

$$\sum_{k|(k,i)\in A} x_{ki} = y_i, \quad i \in V \setminus \{1\}, \tag{10}$$

$$\sum_{(i,j)\in A | i \in S, j \in V \setminus S} x_{ij} \geq y_k, \quad S \subset V, 1 \in S, k \in V \setminus S, \tag{11}$$

$$\sum_{(i,j)\in A} c_{ij} x_{ij} \leq b, \tag{12}$$

$$\sum_{t=2}^{\ell} x_{i_{t-1},i_t} \leq h, \quad P = (i_1 = 1, \ldots, i_\ell) \in \mathcal{P}_h, \tag{13}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in A, \tag{14}$$

$$y_1 = 1, \tag{15}$$

$$y_i \in \{0,1\}, \quad i \in V \setminus \{1\}. \tag{16}$$

Constraints (10) guarantee that if a vertex is selected, it has an indegree of one and vice versa. Constraints (11) guarantee that the selected vertices are connected and are therefore called *connectivity constraints*. Note that because constraints (11) have a variable right-hand side, connectivity does not mean that the solution will span all vertices, but only that the solution will be connected. Also note that in this case variables y_i could be written in terms of the variables x_{ij}, but they are kept in the model for clarity. Fischetti [14] has shown that these constraints can be rewritten as a directed version of the subtour elimination constraints (3). A number of studies have shown that for several variants of the STP and MSTP, directed models are better than their undirected counterpart (see, e.g., [4, 5, 13, 31, 34]). For this reason, we prefer model (9)–(16) to model (1)–(8).

2.2. Miller-Tucker-Zemlin Formulation

A different directed formulation can be obtained by adapting the MTZ constraints. Gouveia [21] has used this idea to propose a formulation for the MST with hop constraints, whereas Khoury et al. [28] have used the same set of constraints in a pure STP formulation. Later, Voß [40] has adapted these formulations to the STP with hop constraints. The basic idea of the MTZ constraints is to introduce potential variables associated with each vertex of the graph and impose that in every path of the solution, the potential variables become larger when the distance from the root vertex increases. Let u_i be a real-valued potential variable associated with vertex i. Again, h is the maximum number of arcs between any vertex and the root vertex. One can model the STPRBH as follows:

Miller-Tucker-Zemlin (MTZ) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{17}$$

subject to

$$\sum_{k|(k,i)\in A} x_{ki} = y_i, \quad i \in V\backslash\{1\}, \tag{18}$$

$$x_{ij} \leq \sum_{k|(k,i)\in A, k\neq j} x_{ki} \quad (i,j) \in A, i \neq 1, \tag{19}$$

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \leq b, \tag{20}$$

$$u_1 = 0, \tag{21}$$

$$y_i \leq u_i \leq hy_i, \quad i \in V\backslash\{1\}, \tag{22}$$

$$(h+1)x_{ij} + u_i - u_j + (h-1)x_{ji} \leq h \quad (i,j) \in A, \tag{23}$$

$$y_1 = 1, \tag{24}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in A, \tag{25}$$

$$y_i \in \{0,1\}, \quad i \in V\backslash\{1\}. \tag{26}$$

Constraints (19) and (21)–(23) guarantee that the solution is connected and cycle-free. Indeed, constraints (23) imply that each vertex in a path has an associated potential variable u_i larger than that of its predecessor, which is impossible in a circuit. Constraints (21)–(22) guarantee that the largest potential in the solution does not exceed h.

Some valid inequalities can be added to the model. For example, for vertices i with zero revenue, constraints (22) can be tightened, as

$$y_i \leq u_i \leq (h-1)y_i \quad i \in V\backslash\{1\}, \;\; r_i = 0, \tag{27}$$

which are valid because there is always an optimal solution containing no unprofitable vertices as leaves. The right-hand side of these constraints can be further strengthened, as proposed by Voß [40], by considering the arcs originating at the root node:

$$y_i \leq u_i \leq (h-1)y_i - (h-2)x_{1i} \quad i \in V\backslash\{1\}, \;\; r_i = 0. \tag{28}$$

The consideration of the arcs emanating from the root enables the strengthening of constraints (22) also for the case of profitable vertices:

$$y_i \leq u_i \leq hy_i - (h-1)x_{1i} \quad i \in V\backslash\{1\}, \;\; r_i > 0. \tag{29}$$

Finally, [40] also proposes several strengthenings of constraints (23), depending on the type of the vertices involved: terminal vertices or Steiner vertices. We can adapt one of these modifications to the case where the two vertices involved in the constraint are unprofitable. In this case, we know that none of the vertices is a leaf and, therefore, the difference between their potentials is bounded by $h - 1$. Constraints (23) then become:

$$hx_{ij} + u_i - u_j + (h-2)x_{ji} \leq h - 1 \quad (i,j) \in A, r_i = r_j = 0. \tag{30}$$

2.3. Garcia-Gouveia Hop Formulation

Garcia [15] and Gouveia [24] have worked with position variables (also called time-indexed variables) to model subtour elimination constraints. The authors mention the possibility of using such variables in situations where a hop limit is imposed. Indeed, let each arc variable x_{ij} be replaced by a set of variables $x_{ij}^p, p = 1, \ldots, h$, where x_{ij}^p is equal to one if arc (i,j) is in position p in the solution, i.e., p hops away from the root, and equal to zero otherwise. In the presence of these disaggregated variables, constraints (19) of the MTZ model can be strengthened and rewritten as

$$x_{ij}^p \leq \sum_{k|(k,i)\in A, k\neq j} x_{ki}^{p-1} \quad (i,j) \in A, \;\; p = 2, \ldots, h, \tag{31}$$

and the STPRBH can be formulated as:

Garcia-Gouveia Hop (GG-Hop) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{32}$$

subject to

$$\sum_{k|(k,i)\in A} x_{ki} = y_i, \quad i \in V\backslash\{1\}, \tag{33}$$

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \leq b, \tag{34}$$

$$x_{ij}^p \leq \sum_{k|(k,i)\in A, k\neq j} x_{ki}^{p-1} \quad (i,j) \in A, i \neq 1, p = 2, \ldots, h. \tag{35}$$

$$x_{ij} = \sum_{p=1}^{h} x_{ij}^p, \quad (i,j) \in A, \tag{36}$$

$$x_{ij}^1 = 0, \quad (i,j) \in A, i \neq 1, \tag{37}$$

$$x_{1j}^p = 0, \quad j \in V\backslash\{1\}, p = 2, \ldots, h, \tag{38}$$

$$x_{ki}^{h} = 0, \quad (k,i) \in A, r_i = 0 \tag{39}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in A. \tag{40}$$

$$y_1 = 1, \tag{41}$$

$$y_i \in \{0,1\}, \quad i \in V\backslash\{1\}. \tag{42}$$

The two-index variables x_{ij} could be rewritten in terms of the three-index variables x_{ij}^{p} only, but they are kept for clarity. The strengthened constraints (35) not only ensure the connectivity of the solution, but also guarantee that the solution is cycle-free and that the hop limit is respected. Note also that constraints (37) and (38) fix some variables based on the fact that no arc originating at the root may be more than one hop away from the root and, conversely, no intermediate arc may be at exactly one hop from the root. Similarly, (39) states that there is always an optimal solution where all leaves are profitable vertices.

In the presence of three-index position variables x_{ij}^{p}, we propose four new families of valid inequalities:

$$\sum_{p=1}^{h} p x_{ij}^{p} \geq \sum_{p=1}^{h} \sum_{k|(k,i)\in A} p x_{ki}^{p} - h + (h+1)x_{ij}, \quad (i,j) \in A, \tag{43}$$

$$x_{ij}^{p} \geq \sum_{k|(k,i)\in A} x_{ki}^{p-1} + x_{ij} - 1,$$
$$(i,j) \in A, i \neq 1, p = 2, \ldots, h, \tag{44}$$

$$\sum_{k|(i,k)\in A} x_{ik}^{p} \geq \sum_{k|(k,i)\in A} x_{ki}^{p-1},$$
$$i \in V, i \neq 1, r_i = 0, p = 2, \ldots, h, \tag{45}$$

$$x_{ij}^{p} \leq 1 - \sum_{k|(k,i)\in A} \sum_{t=p}^{h} x_{ki}^{t} - \sum_{t=1}^{p-1} x_{ij}^{t}$$
$$(i,j) \in A, i \neq 1, p = 2, \ldots, h. \tag{46}$$

Inequalities (43) are a linearization of the constraints

$$\sum_{p=1}^{h} p x_{ij}^{p} \geq \left(\sum_{p=1}^{h} \sum_{k|(k,i)\in A} p x_{ki}^{p} + 1 \right) x_{ij}, \quad (i,j) \in A, \tag{47}$$

which indicate that the hop associated with a given arc depends on the number $\sum_{p=1}^{h} \sum_{k|(k,i)\in A} p x_{ki}^{p}$ of arcs in the path between the root and its origin vertex. In other words, any arc (i,j) leaving vertex i is one more hop away from the root than the arc entering vertex i. Inequalities (44) mean that if arc (i,j) is in the solution, its position should exceed that of its predecessor by one. Inequalities (45) state that no unprofitable vertex i is a leaf in the solution. These are a three-index version of the valid inequalities

$$\sum_{k|(i,k)\in A} x_{ik} \geq \sum_{k|(k,i)\in A} x_{ki}, \quad i \in V, i \neq 1, r_i = 0. \tag{48}$$

Finally, inequalities (46) state that an arc cannot be p hops away from the root if its origin vertex has any incoming arc, which is at least p hops away from the root or if another three-index variable associated with the arc is already set to one.

Gouveia [24] has proved that constraints (35), in their integer version, imply both the subtour and the hop constraints, but he provides no theoretical comparison between the different LP models. The following results prove that the GG-Hop formulation implies the MTZ formulation. More precisely, they show that the MTZ model constraints (19), (22), and (23) are redundant for the GG-Hop model in the presence of the linking constraints

$$u_i = \sum_{p=1}^{h} \sum_{k|(k,i)\in A} p x_{ki}^{p}. \quad i \in V\backslash\{1\}. \tag{49}$$

Note that these constraints only define the u_i variables and do not strengthen the original GG-Hop formulation.

Proposition 1. *Constraints (19) are redundant for the GG-Hop formulation.*

Proof. For any arc $(i,j) \in A, i \neq 1$, summing up the $h-1$ constraints (35) associated with the arc yields:

$$\sum_{p=2}^{h} x_{ij}^{p} \leq \sum_{p=2}^{h} \sum_{k|(k,i)\in A, k\neq j} x_{ki}^{p-1}.$$

Because $i \neq 1$, it follows that $x_{ij}^{1} = 0$ and therefore, $\sum_{p=2}^{h} x_{ij}^{p} = \sum_{p=1}^{h} x_{ij}^{p} = x_{ij}$. This implies

$$x_{ij} \leq \sum_{p=2}^{h} \sum_{k|(k,i)\in A, k\neq j} x_{ki}^{p-1} \leq \sum_{k|(k,i)\in A, k\neq j} x_{ki}.$$

■

Proposition 2. *Constraints (23) are redundant for the GG-Hop formulation.*

Proof. We prove that constraints (23) are redundant for the GG-Hop formulation by showing that in the presence of the disaggregated variables and constraints (35), the left-hand side of any of these constraints is smaller than h. Consider the original MTZ constraint (23) associated with arc (i,j):

$$(h+1)x_{ij} + u_i - u_j + (h-1)x_{ji} \leq h.$$

If vertex i is the root vertex, the result follows easily because $u_i = u_1 = 0$ and $(j,1) \notin A$. Moreover, using (36) and (49), the left-hand side of (23) becomes

$$L = (h+1) \sum_{p=1}^{h} x_{1j}^{p} - \sum_{p=1}^{h} \sum_{k|(k,j)\in A} p x_{kj}^{p}$$
$$= (h+1) \sum_{p=1}^{h} x_{1j}^{p} - \sum_{p=1}^{h} p x_{1j}^{p} - \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq 1} p x_{kj}^{p},$$
$$= \sum_{p=1}^{h} (h+1-p) x_{1j}^{p} - \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq 1} p x_{kj}^{p} \leq h.$$

If i is not the root vertex, we again use (49) to write

$$u_i = \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p, \quad \text{and}$$

$$u_j = \sum_{p=1}^{h} px_{ij}^p + \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p. \quad (50)$$

Using (36) and (50), we can rewrite the left-hand side of (23) as

$$\begin{aligned} L &= (h+1)x_{ij} + u_i - u_j + (h-1)x_{ji} \\ &= (h+1)\sum_{p=1}^{h} x_{ij}^p + \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p - \sum_{p=1}^{h} px_{ij}^p \\ &\quad - \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p + (h-1)\sum_{p=1}^{h} x_{ji}^p. \end{aligned}$$

Defining $a = -\sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p + (h-1)\sum_{p=1}^{h} x_{ji}^p$, we can write

$$L = \sum_{p=1}^{h} (h+1-p)x_{ij}^p + \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p + a. \quad (51)$$

Constraints (35) and (51) imply

$$\begin{aligned} L &\leq hx_{ij}^1 + \sum_{p=1}^{h-1} \sum_{k|(k,i)\in A, k\neq j} (h-p)x_{ki}^p + \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p + a \\ &= hx_{ij}^1 + h\sum_{k|(k,i)\in A} x_{ki}^h + h\sum_{p=1}^{h-1} \sum_{k|(k,i)\in A} x_{ki}^p \\ &\quad - \sum_{p=1}^{h-1} (h-p)x_{ji}^p + a \\ &= hx_{ij}^1 + h\sum_{p=1}^{h} \sum_{k|(k,i)\in A} x_{ki}^p - \sum_{p=1}^{h-1} (h-p)x_{ji}^p + a. \end{aligned}$$

We now show that $-\sum_{p=1}^{h-1}(h-p)x_{ji}^p + a \leq 0$. Again, we use (35) to obtain an upper bound:

$$\begin{aligned} &-\sum_{p=1}^{h-1} (h-p)x_{ji}^p + a \\ &= -\sum_{p=1}^{h-1} (h-p)x_{ji}^p - \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p + (h-1)\sum_{p=1}^{h} x_{ji}^p, \\ &= \sum_{p=1}^{h} (p-1)x_{ji}^p - \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p \\ &\leq \sum_{p=1}^{h-1} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p - \sum_{p=1}^{h} \sum_{k|(k,j)\in A, k\neq i} px_{kj}^p \\ &= -\sum_{k|(k,j)\in A, k\neq i} hx_{kj}^h \leq 0. \end{aligned}$$

It follows that

$$L \leq hx_{ij}^1 + h\sum_{p=1}^{h} \sum_{k|(k,i)\in A} x_{ki}^p \leq h,$$

which proves our proposition since (37) implies $hx_{ij}^1 = 0$ and (33) implies

$$h\sum_{p=1}^{h} \sum_{k|(k,i)\in A} x_{ki}^p = h\sum_{k|(k,i)\in A} \sum_{p=1}^{h} x_{ki}^p = h\sum_{k|(k,i)\in A} x_{ki} = hy_i \leq h.$$

∎

Proposition 3. *Constraints (28) and (29) are redundant for the GG-Hop formulation.*

Proof. Constraints (33), (36), (38), (39), and (49) imply (28) because, for any unprofitable vertex i,

$$\begin{aligned} u_i &= \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p = \sum_{p=1}^{h} \sum_{k|(k,i), k\neq 1\in A} px_{ki}^p + x_{1i}^1 \\ &= \sum_{p=1}^{h} \sum_{k|(k,i)\in A, k\neq 1} px_{ki}^p + (h-1)x_{1i}^1 - (h-2)x_{1i}^1 \\ &\leq (h-1)\sum_{p=1}^{h} \sum_{k|(k,i)\in A, k\neq 1} x_{ki}^p + (h-1)x_{1i}^1 - (h-2)x_{1i}^1 \\ &= (h-1)\sum_{p=1}^{h} \sum_{k|(k,i)\in A} x_{ki}^p - (h-2)x_{1i} \\ &= (h-1)y_i - (h-2)x_{1i} \end{aligned}$$

and,

$$u_i = \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p \geq \sum_{p=1}^{h} \sum_{k|(k,i)\in A} x_{ki}^p = \sum_{k|(k,i)\in A} x_{ki} = y_i.$$

Similar arguments can be used to show that (29) is also redundant for GG-Hop:

$$\begin{aligned} u_i &= \sum_{p=1}^{h} \sum_{k|(k,i)\in A} px_{ki}^p = \sum_{p=1}^{h} \sum_{k|(k,i), k\neq 1\in A} px_{ki}^p + x_{1i}^1 \\ &= \sum_{p=1}^{h} \sum_{k|(k,i)\in A, k\neq 1} px_{ki}^p + hx_{1i}^1 - (h-1)x_{1i}^1 \end{aligned}$$

$$\leq h\sum_{p=1}^{h}\sum_{k|(k,i)\in A,k\neq 1} x_{ki}^p + hx_{1i}^1 - (h-1)x_{1i}^1$$

$$= h\sum_{p=1}^{h}\sum_{k|(k,i)\in A} x_{ki}^p - (h-1)x_{1i} = hy_i - (h-1)x_{1i}.$$

■

Proposition 4. *The GG-Hop model dominates the MTZ model.*

Proof. The result follows from Propositions 1, 2, and 3 and from the fact that constraints (28) and (29) are liftings of (22). ■

Finally, we can use similar arguments to prove that the MTZ model with the lifted constraints (30) is also implied by the GG-Hop formulation, as shown by the following proposition.

Proposition 5. *Constraints (30) are redundant for the GG-Hop formulation.*

Proof. The arguments are very similar to those used to prove Proposition 2. Here, we also use (39), which states that for any unprofitable vertex $i, x_{ik}^h = 0, (i,k) \in A$. We prove that the left-hand side of (30) associated with an arc (i,j) cannot exceed $h-1$ in the presence of the three-index variables x_{ij}^p and the associated constraints. Again, we consider two cases. If i is the root vertex, then the left-hand side L of (30) can be written as

$$\begin{aligned} L &= hx_{1j} - u_j \\ &= h\sum_{p=1}^{h} x_{1j}^p - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq 1} px_{kj}^p - \sum_{p=1}^{h} px_{1j}^p \\ &= \sum_{p=1}^{h}(h-p)x_{1j}^p - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq 1} px_{kj}^p \\ &\leq (h-p)x_{1j} \leq (h-1). \end{aligned}$$

Otherwise, we know that $x_{ik}^1 = 0$ and therefore,

$$\begin{aligned} L &= hx_{ij} + u_i - u_j + (h-2)x_{ji} \\ &= h\sum_{p=1}^{h} x_{ij}^p + \sum_{p=1}^{h}\sum_{k|(k,i)\in A,k\neq j} px_{ki}^p + \sum_{p=1}^{h} px_{ji}^p \\ &\quad - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p - \sum_{p=1}^{h} px_{ij}^p + (h-2)\sum_{p=1}^{h} x_{ji}^p \\ &= \sum_{p=1}^{h}(h-p)x_{ij}^p + \sum_{p=1}^{h}\sum_{k|(k,i)\in A,k\neq j} px_{ki}^p \\ &\quad + \sum_{p=1}^{h}(h+p-2)x_{ji}^p - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p \\ &\leq \sum_{p=1}^{h}\sum_{k|(k,i)\in A,k\neq j}(h-p-1)x_{ki}^p + \sum_{p=1}^{h}\sum_{k|(k,i)\in A,k\neq j} px_{ki}^p \\ &\quad + \sum_{p=1}^{h}(h+p-2)x_{ji}^p - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p \\ &= \sum_{p=1}^{h}\sum_{k|(k,i)\in A,k\neq j}(h-1)x_{ki}^p + \sum_{p=1}^{h}(h+p-2)x_{ji}^p \\ &\quad - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p \\ &= \sum_{p=1}^{h}\sum_{k|(k,i)\in A}(h-1)x_{ki}^p + \sum_{p=1}^{h}(p-1)x_{ji}^p \\ &\quad - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p \\ &\leq \sum_{p=1}^{h}\sum_{k|(k,i)\in A}(h-1)x_{ki}^p + \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p \\ &\quad - \sum_{p=1}^{h}\sum_{k|(k,j)\in A,k\neq i} px_{kj}^p \\ &= \sum_{p=1}^{h}\sum_{k|(k,i)\in A}(h-1)x_{ki}^p \leq (h-1)\sum_{k|(k,i)\in A} x_{ki} \leq h-1. \end{aligned}$$

■

3. BRANCH-AND-CUT ALGORITHMS

We have presented four mathematical formulations for the STPRBH, based on quite different ideas. Models UDFJ and DDFJ use the classical Dantzig-Fulkerson-Johnson subtour elimination constraints, while MTZ model uses the Miller-Tucker-Zemlin constraints, and GG-Hop is based on three-index position variables. We have shown that model MTZ and its lifted version are dominated by GG-Hop and, therefore, they will not be considered in the tests. Moreover, as we have explained in Section 2.1, we concentrate on the directed DDFJ formulation.

Because of the exponential number of constraints of DDFJ, it is impractical to apply commercial solvers to this formulation, even for mid-size instances. We propose a branch-and-cut algorithm containing two separation procedures which generate violated constraints of type (11) and (13) only when these are found to be violated during the branch-and-cut process. Section 3.1 explains the two separation procedures and the overall branch-and-cut algorithms.

The number of constraints in formulations GG-Hop is polynomial and, therefore, one can make use of commercial LP solvers for the problem as long as the instance size is not too large. However, it is possible to relax constraints (35)

TABLE 1. Summary of the branch-and-cut strategies.

Strategy	Basic model	Relaxed constraints	Valid inequalities
$S1$	GG-Hop		
$S2$	GG-Hop		(11), (13)
$S3$	GG-Hop	(35)	
$S4$	GG-Hop	(35)	(11), (13)
$S5$	DDFJ + (52), (53)	(11), (13)	

and add them as cuts in the model when they are found to be violated during the branching process. Section 3.2 presents this second branch-and-cut approach.

3.1. Branch-and-Cut for Formulation DDFJ

We consider the DDFJ model. The idea of our branch-and-cut algorithm is to initially relax connectivity constraints (11) and hop constraints (13) and add them as cuts only when they become violated. To reinforce the formulation at the root node, we do not relax all of constraints (11). Instead, the following constraints are kept:

$$x_{ij} \leq \sum_{k|(k,i)\in A, k\neq j} x_{ki} \quad (i,j) \in A, i \neq 1. \tag{52}$$

Algorithm 1. Hop separation procedure.

$\ell = 1; P = (i_\ell = 1); X_P = 0; f = [0, 1, 1, \ldots, 1];$
loop
  if there exists $(i_\ell, j) \in A$, with $f[j] = 1$ **then**
    $\ell = \ell + 1;$
    $i_\ell = j;$
    $f[i_\ell] = 0;$
    $X_P = X_P + x_{i_{\ell-1},i_\ell};$
    if $\ell = h + 2$ and $X_P > h$ **then**
      Add the cut $X_P \leq h$;
      $f[k] = 1$, for all k such that $(i_\ell, k) \in A, k \neq i_{\ell-1}$;
      $X_P = X_P - x_{i_{\ell-1},i_\ell};$
      $\ell = \ell - 1;$
    else
      if $X_P < \ell - 2$ **then**
        $X_P = X_P - x_{i_{\ell-1},i_\ell};$
        $\ell = \ell - 1;$
      end if
    end if
  else
    if $\ell = 1$ **then**
      Stop.
    else
      $X_P = X_P - x_{i_{\ell-1},i_\ell};$
      $f[k] = 1$, for all k such that $(i_\ell, k) \in A, k \neq i_{\ell-1}$;
      $\ell = \ell - 1;$
    end if
  end if
end loop

Algorithm 2. Branch-and-cut template.

while constraints added **do**
  Solve root node.
  Look for violated relaxed constraints.
  Look for violated valid inequalities.
end while
if solution is integer **then**
  Stop.
else
  Start branching.
  while there exist active nodes **do**
    Branch or change current node.
    while constraints added **do**
      Solve current node.
      Look for violated relaxed constraints.
      Look for violated valid inequalities.
    end while
  end while
end if

Note that constraints (52) are equivalent to constraints (11) associated with sets $S = \{i,j\}$, for which $(i,j) \in A$. Indeed, these constraints can be obtained by adding the term $\sum_{(p\neq i,j)\in A} x_{pj}$ to both sides of (52). In addition to these constraints, we also explicitly impose the two-vertex subtour elimination constraints, which are implied by (10) and (11):

$$x_{ij} + x_{ji} \leq 1 \quad (i,j) \in E. \tag{53}$$

For the relaxed constraints, it is necessary to develop appropriate separation procedures. The first separation procedure takes care of the connectivity of the solution. This procedure is adapted from the work of Ljubić et al. [31]. It consists of identifying disconnected vertices by means of a maximum flow algorithm, and of adding the associated violated constraints of type (11). The second separation procedure identifies violated hop constraints of type (13) by finding paths from the root to any vertex containing more

TABLE 2. Optimal solutions for instance *Steinb*8 for the 12 different scenarios.

Scenario	Budget	Cost	h	Optimal solution value	Profitable vertices reached	% Revenue collected	Figure
1	501	11	3	85	2	10	1a
2	501	114	6	581	13	70	1b
3	501	167	9	836	19	100	1c
4	501	178	12	836	19	100	1c
5	100.2	11	3	85	2	10	1a
6	100.2	98	6	535	11	64	1d
7	100.2	100	9	761	16	91	1e
8	100.2	100	12	832	18	99	2a
9	50.1	11	3	85	2	10	1a
10	50.1	48	6	346	6	41	2b
11	50.1	50	9	537	10	64	2c
12	50.1	50	12	537	10	64	2c

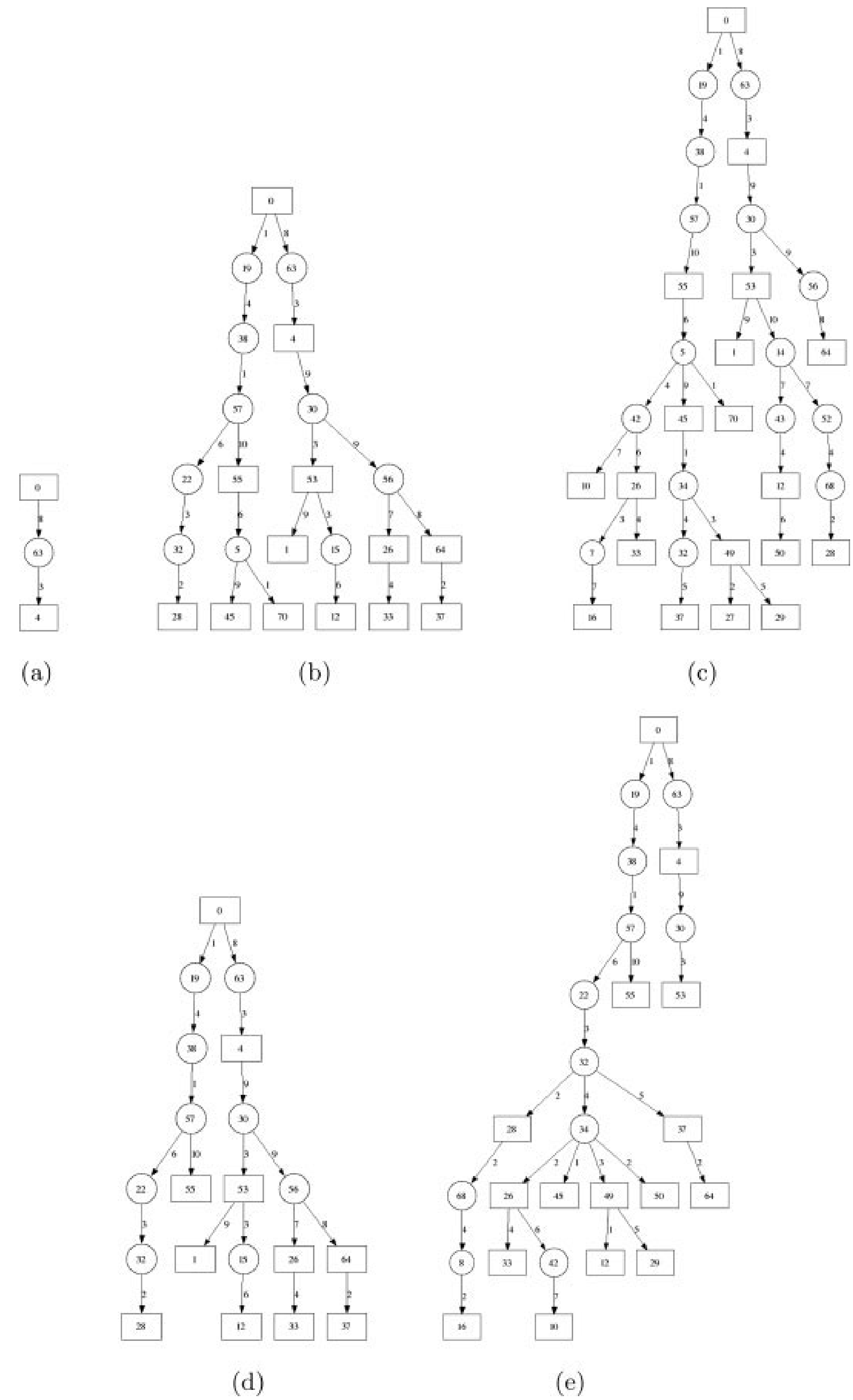

FIG. 1. Solutions for instance *Steinb8* in the different scenarios. (a) Scenarios 1, 5, 9, (b) Scenario 2, (c) Scenarios 3, 4, (d) Scenario 6, and (e) Scenario 7.

arcs than the allowed hop limit. These two routines are now presented in detail.

3.1.1. Connectivity Separation Procedure. The connectivity separation procedure exploits the fact that constraints (11) imply the connectivity of all selected vertices in the graph, in particular, the connectivity of the root to all other selected vertices. Indeed, given a partition $\{S, V\backslash S\}$ of V and a cut $C = \{(i,j) : i \in S, j \in V\backslash S\}$ in which the root and a given vertex k are on different sides of the cut, i.e., $1 \in S$ and $k \in V\backslash S$, let X_C be the sum of the x variables belonging to the cut: $X_C = \sum_{(i,j)\in C} x_{ij}$. The connectivity constraints guarantee that $X_C \geq y_k$. Now, if one uses the current x values at a given node of the branching tree as arc capacities, it is possible to use a maximum flow algorithm to find the minimum cut, in terms of X_C, between the root and every other

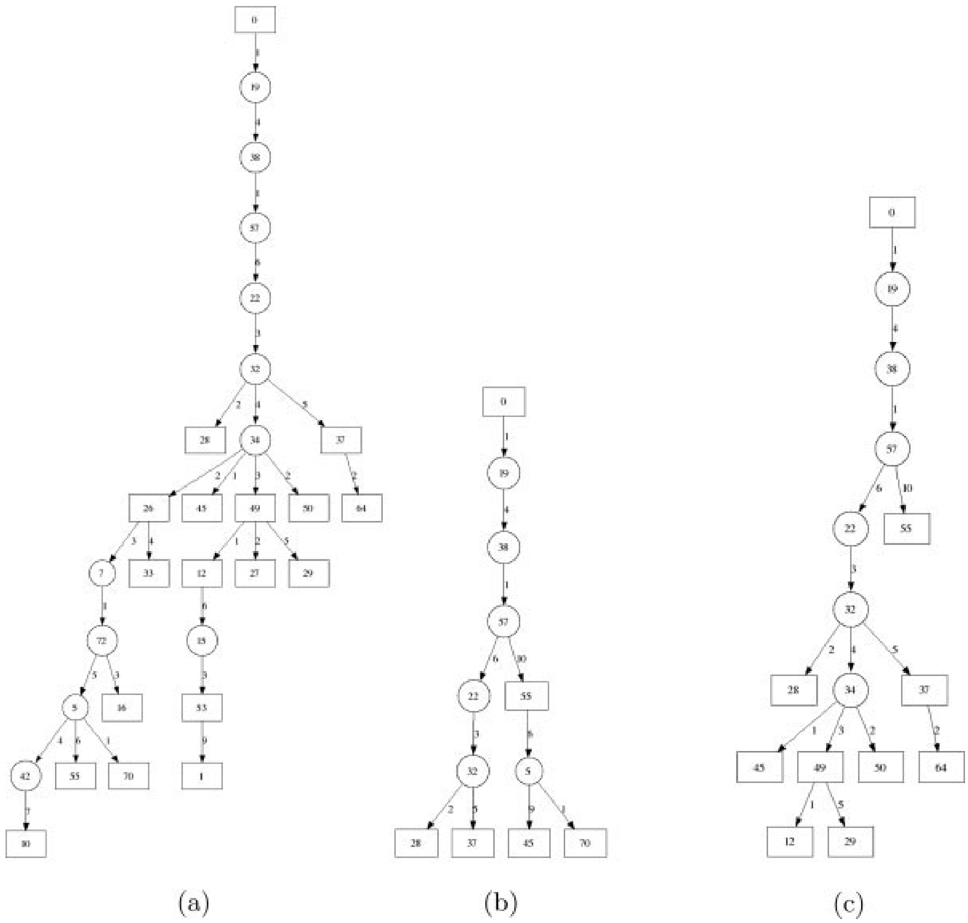

FIG. 2. Solutions for instance *Steinb8* in the different scenarios (continued). (a) Scenario 8, (b) Scenario 10, and (c) Scenarios 11, 12.

vertex k of the graph. If X_C is smaller than y_k, then a violated connectivity constraint has been identified and can be added to the problem.

To identify more violated inequalities at each iteration, Ljubić et al. [31] have proposed the use of *nested cuts* and *back-cuts*. The nested cuts were introduced by Martin [30]. Once a violated cut is found, these cuts consist of setting to one the values of the x variables in the cut and of solving the minimum cut problem again, in the hope of finding another cut that does not include all of the previous arcs. The process is repeated until no more cuts can be identified or an upper limit is reached. The back-cuts were introduced by Chopra and Rao [4]. The idea is simply to reverse the flows and look for a minimum cut with i as origin vertex and the root vertex as destination. In our implementation, as suggested by Ljubić et al. [31], the nested cuts and the back-cuts are combined, maximizing the number of violated constraints found at each iteration.

3.1.2. Hop Constraints Separation Procedure. The search for violated hop constraints consists of exploring the current tree for paths containing more arcs than the hop limit. We concentrate on paths rooted at vertex 1 and we explore the subgraph induced by the positive y_i and x_{ij} variables. For a path $P = (i_1 = 1, \ldots, i_\ell)$, define $X_P = \sum_{t=2}^{\ell} x_{i_{t-1},i_t}$ as the sum of the x_{ij} variables in the path. The separation procedure gradually extends a path P rooted at 1 until the number of vertices in the path reaches $h + 2$ and the sum of associated variables exceeds h, or this sum is less than the number of vertices minus two. In other words, these conditions are:

1. $\ell = h + 2$ and $X_P > h$,
2. $X_P < \ell - 2$.

In condition 1 a violated hop constraint has been found and the cut $X_P \leq h$ is added to the model. In condition 2 it is useless to continue exploring the branch, because no violated inequalities will be found. The complete separation procedure is described in Algorithm 1, where $f[i] = 1$ if vertex i can still been visited in the search ($i = 1, \ldots, n$), and 0 otherwise.

The idea of Algorithm 1 is to sequentially visit the connected vertices in the induced subgraph until one of conditions 1 or 2 is met. Again, if condition 1 is satisfied, a violated cut has been identified and is added to the model. In both cases, the algorithm blocks the access to the last vertex in the path and returns to the previous vertex, from which the search is resumed. The procedure continues until the root vertex is reached, with no other vertices to visit.

TABLE 3. Results - instances MStein, $h = 3$.

Instance					$S1$		$S3$			$S5$			
Name	n	$\|E\|$	b	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds
Msteinb1	50	63	1	140	0	**0.01**	0	102 (40.48%)	0.03	279	4,245	744	1.77
			5	140	0	**0.01**	0	78 (30.95%)	0.04	816	2,718	666	2.56
			10	140	0	**0.01**	0	78 (30.95%)	0.04	178	283	100	0.31
Msteinb2	50	63	1	182	0	**0.01**	0	123 (48.81%)	**0.01**	445	4,902	1,738	5.79
			5	182	0	**0.01**	0	126 (50.00%)	0.05	247	1,724	740	2.36
			10	182	0	**0.01**	4	118 (46.83%)	0.07	72	508	206	0.53
Msteinb3	50	63	1	253	0	**0.01**	0	163 (64.68%)	0.02	9	268	194	0.3
			5	253	0	**0.01**	0	154 (61.11%)	0.05	4	174	132	0.25
			10	253	0	**0.01**	0	140 (55.56%)	0.05	4	151	104	0.18
Msteinb4	50	100	1	341	0	**0.02**	1	199 (49.75%)	0.07	15,609	486,329	32,432	[inf]
			5	341	0	**0.01**	130	252 (63.00%)	0.53	17,751	154,921	14,198	[inf]
			10	341	0	**0.01**	110	151 (37.75%)	0.81	26,908	147,184	17,087	[16.83%]
Msteinb5	50	100	1	588	0	**0.02**	0	153 (38.25%)	0.05	24,853	740,421	32,770	[inf]
			5	588	0	**0.02**	72	142 (35.50%)	0.36	19,146	281,447	17,530	[inf]
			10	565	0	**0.02**	0	58 (14.50%)	0.05	38,848	109,378	10,819	4,416.86
Msteinb6	50	100	1	1,040	0	**0.02**	11	250 (62.50%)	0.11	23,816	534,587	31,211	[inf]
			5	1,035	0	**0.01**	68	163 (40.75%)	0.33	24,005	365,437	31,894	[31.84%]
			10	610	10	**0.04**	120	152 (38.00%)	0.51	41,167	78,719	9,904	4,432.77
Msteinb7	75	94	1	87	0	**0.02**	0	211 (56.12%)	0.04	28	1,279	528	2.08
			5	87	0	**0.01**	4	183 (48.67%)	0.09	4	657	292	0.84
			10	87	0	**0.01**	78	174 (46.28%)	0.29	6	421	210	0.5
Msteinb8	75	94	1	85	0	**0.01**	0	206 (54.79%)	0.04	116	2,154	830	3.31
			5	85	0	**0.01**	0	197 (52.39%)	0.08	40	970	570	2.72
			10	85	0	**0.01**	8	176 (46.81%)	0.12	16	324	238	0.75
Msteinb9	75	94	1	596	0	**0.01**	0	231 (61.44%)	0.03	9,521	262,941	42,102	4,228.27
			5	596	0	**0.01**	2	190 (50.53%)	0.11	7,566	73,353	15,871	1,087.73
			10	483	3	**0.02**	80	175 (46.54%)	0.29	1,132	7,056	2,278	19.49
Msteinb10	75	150	1	319	0	**0.02**	10	359 (59.83%)	0.18	7,115	1,409,190	47,418	[inf]
			5	319	0	**0.02**	391	388 (64.67%)	2.04	6,383	432,980	22,686	[inf]
			10	319	0	**0.02**	125	275 (45.83%)	1.48	14,440	304,253	22,341	[64.47%]
Msteinb11	75	150	1	316	0	**0.02**	4	393 (65.50%)	0.15	6,893	1,100,426	45,110	[inf]
			5	316	0	**0.02**	26	352 (58.67%)	0.46	6,737	353,237	25,042	[inf]
			10	305	0	**0.02**	213	329 (54.83%)	3.07	6,102	178,547	21,538	[76.02%]
Msteinb12	75	150	1	1,169	0	**0.02**	0	418 (69.67%)	0.09	6,015	906,773	35,515	[inf]
			5	1,169	0	**0.02**	78	349 (58.17%)	0.7	6,734	664,978	41,020	[inf]
			10	1,017	1	**0.03**	116	202 (33.67%)	0.71	12,710	335,980	28,226	[inf]
Msteinb13	100	125	1	147	0	**0.02**	0	241 (48.20%)	0.06	2,171	377,891	51,584	[81.27%]
			5	147	0	**0.02**	47	234 (46.80%)	0.39	6,306	183,147	30,340	[75.94%]
			10	147	0	**0.02**	46	175 (35.00%)	0.51	2,386	18,519	4,848	162.37
Msteinb14	100	125	1	263	0	**0.02**	0	272 (54.40%)	0.07	189	14,482	3,400	54.13
			5	263	0	**0.02**	0	252 (50.40%)	0.14	872	12,783	4,146	95.79
			10	263	0	**0.01**	42	253 (50.60%)	0.56	162	4,380	1,538	15.09
Msteinb15	100	125	1	1,061	0	**0.02**	0	280 (56.00%)	0.05	1,639	709,689	68,728	[61.41%]
			5	1,061	0	**0.01**	0	247 (49.40%)	0.17	26,630	121,736	25,932	[33.34%]
			10	830	0	**0.02**	51	173 (34.60%)	0.55	1,491	6,735	2,132	28.03
Msteinb16	100	200	1	479	0	**0.03**	0	473 (59.13%)	0.19	3,196	1,048,853	38,092	[inf]
			5	479	0	**0.02**	28	460 (57.50%)	0.65	5,941	232,061	13,442	[inf]
			10	479	0	**0.03**	495	522 (65.25%)	7.34	4,118	177,306	18,746	[inf]
Msteinb17	100	200	1	254	0	**0.03**	0	602 (75.25%)	0.2	1,875	126,786	12,748	[inf]
			5	254	0	**0.02**	28	516 (64.50%)	0.74	4,805	121,313	13,210	[inf]
			10	254	0	**0.03**	0	464 (58.00%)	0.47	3,193	178,010	18,231	[76.62%]
Msteinb18	100	200	1	1,298	0	**0.03**	0	542 (67.75%)	0.15	6,331	1,334,390	43,226	[inf]
			5	1,298	0	**0.03**	5	454 (56.75%)	0.37	3,254	540,237	30,572	[inf]
			10	1,132	2	**0.04**	522	461 (57.63%)	3.37	3,424	563,325	34,466	[79.32%]

3.2. Branch-and-Cut for Formulation GG-Hop

Several algorithms can be used to solve the GG-Hop formulation. The simplest approach is to solve GG-Hop directly with a commercial branch-and-cut solver. Variants of this approach are obtained by relaxing constraints (35) at the root of the search tree and dynamically generating these constraints whenever they are found to be violated. Similarly, valid inequalities (11) and (13) can be generated in the search tree. We have also tested the possibility of including (43)–(46)

TABLE 4. Results - instances MStein, $h = 6$.

Instance					S1		S3			S5			
Name	n	$\|E\|$	b	Opt	B& B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds
Msteinb1	50	63	1	403	0	**0.02**	0	191 (30.32%)	0.05	4,351	3,026	598	8.15
			5	403	0	**0.02**	23	57 (9.05%)	0.07	102	182	78	0.32
			10	341	0	0.02	0	23 (3.65%)	0.02	0	0	0	**0.01**
Msteinb2	50	63	1	612	0	**0.01**	2	359 (56.98%)	0.33	14,483	8,200	2,004	53.78
			5	491	7	**0.04**	243	304 (48.25%)	0.68	436	578	296	1.45
			10	300	9	**0.04**	91	118 (18.73%)	0.21	65	6	62	0.24
Msteinb3	50	63	1	963	0	**0.01**	0	401 (63.65%)	0.22	5,256	9,587	1,464	18.7
			5	807	0	**0.02**	170	424 (67.30%)	0.64	720	1,154	296	1.75
			10	624	0	**0.02**	4	85 (13.49%)	0.11	11	45	36	0.1
Msteinb4	50	100	1	455	5	0.19	0	264 (26.40%)	**0.13**	20	1,181	88	0.33
			5	455	10	**0.22**	125	332 (33.20%)	0.98	939	11,760	702	6.92
			10	447	54	**0.69**	163	255 (25.50%)	1.04	546	4,410	496	11.35
Msteinb5	50	100	1	666	0	**0.05**	2	232 (23.20%)	0.14	31	2,104	90	0.53
			5	666	3	**0.24**	98	352 (35.20%)	1	16	1,116	58	0.38
			10	643	4	0.13	25	85 (8.50%)	**0.12**	73	237	26	0.21
Msteinb6	50	100	1	1,269	4	0.39	11	480 (48.00%)	**0.27**	26	5,029	210	0.87
			5	1,257	18	**0.59**	94	358 (35.80%)	0.87	33,813	57,648	3,696	740.81
			10	839	36	**0.65**	237	267 (26.70%)	1.04	1,392	5,325	538	14.19
Msteinb7	75	94	1	627	0	**0.02**	15	572 (60.85%)	0.33	27,591	115,272	21,226	[6.97%]
			5	627	0	**0.02**	30	224 (23.83%)	0.3	195	706	330	1.69
			10	432	2	**0.03**	212	219 (23.30%)	0.6	43	70	80	0.38
Msteinb8	75	94	1	581	0	**0.02**	0	571 (60.74%)	0.79	34,992	128,342	23,778	6,612.82
			5	535	0	**0.02**	169	514 (54.68%)	1.57	31,888	39,172	9,957	2,307.75
			10	346	0	**0.03**	344	494 (52.55%)	2.16	507	2,076	798	4.9
Msteinb9	75	94	1	1,698	0	**0.02**	0	497 (52.87%)	0.18	82,814	238,660	26,620	[3.57%]
			5	1,343	0	**0.04**	39	240 (25.53%)	0.35	1,787	4,997	1,478	17.47
			10	816	12	**0.1**	210	235 (25.00%)	0.66	66	24	22	0.3
Msteinb10	75	150	1	702	3	0.45	4	508 (33.87%)	**0.33**	15,377	1,993,727	42,816	[inf]
			5	702	7	**0.68**	80	511 (34.07%)	1.67	749	10,127	424	9.69
			10	668	77	**1.2**	289	318 (21.20%)	2.17	1,636	13,036	694	34.24
Msteinb11	75	150	1	893	4	**0.18**	0	430 (28.67%)	0.19	16,751	1,384,101	34,858	[inf]
			5	893	8	**0.35**	583	727 (48.47%)	9.23	14,482	340,611	18,360	[inf]
			10	829	3	**0.54**	619	625 (41.67%)	7.41	2,007	22,939	2,330	80.31
Msteinb12	75	150	1	1,867	2	**0.24**	0	663 (44.20%)	0.28	18,908	1,464,622	36,796	[inf]
			5	1,847	20	**0.76**	278	606 (40.40%)	3.54	10,881	1,105,823	42,612	[inf]
			10	1,384	3	**0.09**	48	215 (14.33%)	0.64	370	11,652	1,132	14.92
Msteinb13	100	125	1	785	0	**0.05**	0	516 (41.28%)	0.18	10,178	187,968	23,580	1,415.72
			5	674	31	**0.14**	378	503 (40.24%)	1.87	16,585	15,337	5,337	696.88
			10	452	8	**0.1**	172	394 (31.52%)	1.2	164	340	164	1.7
Msteinb14	100	125	1	1,296	0	**0.02**	4	773 (61.84%)	1.18	27,541	250,445	27,430	[19.67%]
			5	977	1	**0.04**	497	525 (42.00%)	2.45	1,026	4,922	1,968	30.07
			10	595	0	**0.04**	19	89 (7.12%)	0.3	0	1	10	0.05
Msteinb15	100	125	1	2,555	0	**0.04**	0	774 (61.92%)	0.38	18,788	457,920	42,774	[inf]
			5	1,858	12	**0.13**	173	230 (18.40%)	1.09	313	1,801	430	4.3
			10	1,040	38	**0.16**	252	297 (23.76%)	1.19	177	50	38	0.81
Msteinb16	100	200	1	840	1	**0.31**	18	1,414 (70.70%)	11.59	5,564	1,900,715	39,986	[inf]
			5	840	15	**0.68**	407	1,082 (54.10%)	13.34	9,278	505,404	17,587	[inf]
			10	767	106	**2.65**	437	805 (40.25%)	8.21	7,428	98,879	6,036	[4.12%]
Msteinb17	100	200	1	1,299	0	**0.06**	19	1,588 (79.40%)	6.58	1,885	1,600,101	46,970	[inf]
			5	1,299	0	**0.06**	400	1,225 (61.25%)	18.37	9,059	930,397	30,941	[inf]
			10	1,091	0	**0.07**	442	819 (40.95%)	9.04	11,986	273,874	24,689	[inf]
Msteinb18	100	200	1	2,585	10	1.42	11	1,088 (54.40%)	**0.94**	12,412	2,463,977	28,620	[inf]
			5	2,575	8	**0.84**	43	581 (29.05%)	1.77	6,870	880,530	24,466	[inf]
			10	1,917	0	**0.09**	160	562 (28.10%)	2.97	13,420	580,741	26,769	[inf]

TABLE 5. Results - instances MStein, $h = 9$

Instance					$S1$		$S3$			$S5$			
Name	n	$\|E\|$	b	Opt	B & B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds
Msteinb1	50	63	1	467	0	**0.03**	5	236 (23.41%)	0.1	13	372	82	0.17
			5	431	17	0.17	3	78 (7.74%)	0.05	2	0	2	**0.03**
			10	341	0	0.03	0	26 (2.58%)	0.03	0	0	0	**0.01**
Msteinb2	50	63	1	696	0	**0.02**	0	394 (39.09%)	0.14	30	934	192	0.45
			5	600	7	0.09	31	228 (22.62%)	0.19	16	6	12	**0.07**
			10	300	56	0.31	59	60 (5.95%)	**0.18**	62	0	64	0.24
Msteinb3	50	63	1	1,205	0	**0.02**	9	733 (72.72%)	3.31	18	689	94	0.23
			5	924	4	**0.08**	87	324 (32.14%)	0.37	10	46	16	0.09
			10	649	0	0.02	0	80 (7.94%)	0.02	0	0	0	**0.01**
Msteinb4	50	100	1	455	1	0.23	18	1,048 (65.50%)	6.65	0	17	2	**0.02**
			5	455	0	0.08	4	219 (13.69%)	0.15	0	13	2	**0.04**
			10	455	0	0.36	2	143 (8.94%)	0.18	2	424	4	**0.08**
Msteinb5	50	100	1	666	3	0.42	0	359 (22.44%)	**0.14**	9	2,460	46	0.36
			5	666	7	1.36	10	382 (23.88%)	0.56	5	1,110	30	**0.2**
			10	652	0	0.09	0	37 (2.31%)	0.05	0	0	0	**0.02**
Msteinb6	50	100	1	1,269	2	0.59	0	391 (24.44%)	0.25	0	479	24	**0.08**
			5	1,262	18	**2.03**	221	630 (39.38%)	2.34	456	5,284	322	10.38
			10	903	7	0.29	4	160 (10.00%)	0.18	5	6	4	**0.05**
Msteinb7	75	94	1	674	2	**0.07**	16	764 (50.80%)	0.55	394	2,127	380	2.66
			5	662	0	0.04	0	202 (13.43%)	0.1	0	2	2	**0.03**
			10	432	21	**0.12**	50	163 (10.84%)	0.25	21	3	18	0.14
Msteinb8	75	94	1	836	0	**0.04**	4	477 (31.72%)	0.27	8,338	35,111	4,342	112.73
			5	761	36	**0.26**	439	783 (52.06%)	5.08	4,182	4,391	1,086	28.87
			10	537	2	0.14	223	431 (28.66%)	1.84	6	115	26	**0.13**
Msteinb9	75	94	1	1,761	2	**0.15**	14	926 (61.57%)	3.23	52,369	134,183	12,568	3,138.36
			5	1,388	0	**0.06**	3	217 (14.43%)	0.12	8	33	10	0.09
			10	816	107	0.86	142	351 (23.34%)	0.61	60	2	14	**0.26**
Msteinb10	75	150	1	702	3	2.14	0	432 (18.00%)	**0.26**	26	5,045	190	1.67
			5	702	5	1.28	89	577 (24.04%)	1.55	21	3,414	108	**1.19**
			10	694	1	0.29	47	217 (9.04%)	0.96	2	27	10	**0.16**
Msteinb11	75	150	1	893	5	**3.1**	34	1,850 (77.08%)	15.63	35,616	1,748,774	24,754	[inf]
			5	893	7	6.47	105	846 (35.25%)	2.75	6	1,501	46	**0.32**
			10	855	4	1.59	213	598 (24.92%)	3.78	1	136	12	**0.11**
Msteinb12	75	150	1	1,867	5	4.12	37	1,537 (64.04%)	4.23	58	3,862	114	**1.47**
			5	1,866	3	**0.9**	16	519 (21.63%)	1.08	98	14,165	396	5.97
			10	1,401	48	1.18	53	247 (10.29%)	**0.46**	168	369	78	1.43
Msteinb13	100	125	1	785	2	**0.17**	13	826 (41.30%)	0.45	242	12,946	1,378	12.25
			5	745	8	0.41	28	213 (10.65%)	0.3	11	8	20	**0.23**
			10	465	69	0.87	162	342 (17.10%)	1.21	70	33	44	**0.54**
Msteinb14	100	125	1	1,403	0	**0.05**	10	1,296 (64.80%)	10.76	18	1,863	176	1.42
			5	1,033	38	**0.44**	120	365 (18.25%)	0.89	128	580	248	2.31
			10	595	27	0.21	118	139 (6.95%)	0.85	0	0	10	**0.05**
Msteinb15	100	125	1	2,555	1	**0.19**	20	1,341 (67.05%)	6.09	1,715	18,306	1,336	26.69
			5	1,891	0	0.06	0	120 (6.00%)	0.06	0	0	0	**0.03**
			10	1,086	138	2.39	202	252 (12.60%)	1.26	31	6	10	**0.21**
Msteinb16	100	200	1	840	5	**3.79**	26	2,045 (63.91%)	36.87	5,209	206,654	2,576	167.4
			5	840	6	3.39	77	841 (26.28%)	2.84	10	5,015	194	**1.82**
			10	800	21	7.4	10	223 (6.97%)	**0.49**	801	33,077	1,494	118.41
Msteinb17	100	200	1	1,299	5	4.4	0	623 (19.47%)	**0.44**	20,732	1,777,216	19,200	[inf]
			5	1,299	14	**4.91**	367	1,614 (50.44%)	21.89	9,364	627,702	17,550	[inf]
			10	1,178	32	**2.34**	986	1,256 (39.25%)	23.74	15,242	115,951	6,176	5,320.1
Msteinb18	100	200	1	2,585	8	**7.77**	31	2,098 (65.56%)	29.19	1,228	174,680	1,846	75.04
			5	2,585	0	0.86	12	482 (15.06%)	**0.72**	2,824	297,448	4,606	1,613
			10	1,997	8	0.99	7	354 (11.06%)	**0.41**	667	28,636	992	35.4

TABLE 6. Results - instances MStein, $h = 12$.

Instance					$S1$		$S3$			$S5$			
Name	n	$\|E\|$	b	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds
Msteinb1	50	63	1	467	0	0.31	1	202 (14.57%)	0.07	3	24	28	**0.06**
			5	431	2	0.14	3	50 (3.61%)	0.05	2	0	2	**0.03**
			10	341	0	0.04	0	27 (1.95%)	0.03	0	0	0	**0.01**
Msteinb2	50	63	1	696	0	**0.04**	8	324 (23.38%)	0.13	2	35	14	0.05
			5	600	30	0.37	51	332 (23.95%)	0.42	13	0	6	**0.05**
			10	300	77	0.77	55	64 (4.62%)	**0.17**	62	0	64	0.24
Msteinb3	50	63	1	1,205	0	**0.03**	2	535 (38.60%)	0.17	50	1,136	166	0.48
			5	931	39	0.45	101	405 (29.22%)	0.55	24	27	8	**0.09**
			10	649	0	0.03	0	124 (8.95%)	0.04	0	0	0	**0.01**
Msteinb4	50	100	1	455	4	2.38	0	393 (17.86%)	0.24	0	2	2	**0.02**
			5	455	18	6.77	58	449 (20.41%)	0.77	0	0	2	**0.03**
			10	455	4	1.9	0	92 (4.18%)	0.11	0	375	4	**0.04**
Msteinb5	50	100	1	666	3	2.82	8	578 (26.27%)	0.59	0	1,005	10	**0.05**
			5	666	6	5.54	0	185 (8.41%)	**0.1**	6	1,066	22	0.2
			10	652	0	0.14	0	84 (3.82%)	0.07	0	0	0	**0.02**
Msteinb6	50	100	1	1,269	3	2.83	6	502 (22.82%)	0.33	0	286	20	**0.07**
			5	1,262	456	21.66	223	687 (31.23%)	2.39	0	1,001	12	**0.07**
			10	903	28	2.11	4	135 (6.14%)	0.14	6	0	4	**0.06**
Msteinb7	75	94	1	674	1	0.39	3	486 (23.50%)	0.17	0	181	60	**0.14**
			5	662	0	0.14	1	276 (13.35%)	0.16	0	0	2	**0.02**
			10	432	56	0.85	109	345 (16.68%)	0.79	21	0	18	**0.14**
Msteinb8	75	94	1	836	3	**0.21**	0	532 (25.73%)	0.23	39	1,474	262	1.05
			5	832	7	0.57	225	673 (32.54%)	2.55	0	118	12	**0.05**
			10	537	25	0.88	628	732 (35.40%)	5.97	11	74	36	**0.15**
Msteinb9	75	94	1	1,761	0	**0.07**	0	747 (36.12%)	0.3	2	26	10	0.09
			5	1,388	4	0.35	42	448 (21.66%)	0.69	44	90	40	**0.33**
			10	816	107	1.2	154	139 (6.72%)	0.43	60	3	14	**0.26**
Msteinb10	75	150	1	702	9	10.65	0	173 (5.24%)	**0.13**	17	7,909	150	1.55
			5	702	14	21.13	13	523 (15.85%)	**0.64**	21	4,675	74	1.21
			10	694	2	3.48	3	176 (5.33%)	0.34	2	1	10	**0.13**
Msteinb11	75	150	1	893	5	8.32	12	834 (25.27%)	**0.92**	12	7,363	186	1.45
			5	893	11	16.9	117	1,044 (31.64%)	3.86	1	131	28	**0.14**
			10	855	44	5.57	544	861 (26.09%)	9.88	2	169	12	**0.1**
Msteinb12	75	150	1	1,867	5	10.86	5	619 (18.76%)	**0.37**	15	10,956	100	1.37
			5	1,866	0	0.58	0	466 (14.12%)	0.28	4	341	8	**0.18**
			10	1,401	58	1.33	69	261 (7.91%)	**0.5**	130	200	26	0.68
Msteinb13	100	125	1	785	3	0.77	7	1,052 (38.25%)	0.74	0	7	2	**0.03**
			5	745	35	1.54	27	312 (11.35%)	**0.31**	26	3	22	0.32
			10	465	96	1.67	382	653 (23.75%)	3.58	55	5	30	**0.41**
Msteinb14	100	125	1	1,403	2	**0.28**	5	917 (33.35%)	0.54	12	979	194	1.38
			5	1,038	52	0.75	73	533 (19.38%)	0.86	65	2	18	**0.37**
			10	595	21	0.39	6	161 (5.85%)	0.39	0	0	10	**0.05**
Msteinb15	100	125	1	2,555	0	**0.16**	0	997 (36.25%)	0.4	10	845	122	1.12
			5	1,891	0	0.09	0	294 (10.69%)	0.12	0	0	0	**0.03**
			10	1,109	2	0.29	3	149 (5.42%)	**0.19**	21	0	8	0.21
Msteinb16	100	200	1	840	1	10.31	5	1,595 (36.25%)	**1.44**	80	42,144	516	9.72
			5	840	7	28.64	116	1,381 (31.39%)	4.8	14	5,972	120	**2.27**
			10	800	7	16.45	114	620 (14.09%)	**3.03**	104	2,622	128	4.09
Msteinb17	100	200	1	1,299	2	10.11	26	2,770 (62.95%)	38.44	69	59,844	352	**9.11**
			5	1,299	6	19.32	112	1,261 (28.66%)	6.73	1	2,420	58	**0.52**
			10	1,225	0	0.29	3	256 (5.82%)	0.27	2	2	0	**0.12**
Msteinb18	100	200	1	2,585	2	16.83	5	1,229 (27.93%)	**0.97**	21	12,678	166	4.09
			5	2,585	4	20.03	83	938 (21.32%)	**3.4**	242	38,828	358	13.77
			10	1,997	122	14.64	302	721 (16.39%)	**5.2**	390	5,088	94	5.29

TABLE 7. Results - instances steinc, $h = 5$.

Instance					$S1$		$S3$			$S5$			
Name	n	$\|E\|$	b	Opt	B& B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds
steinc1_10	500	625	1	8	0	**0.11**	0	2,959 (59.18%)	34.76	40	277,910	22,912	[inf]
steinc1_100			1	71	0	**0.12**	0	2,959 (59.18%)	34.67	40	277,910	22,912	[inf]
steinc1_10			10	8	0	**0.12**	998	3,121 (62.42%)	186.06	362	161,716	23,259	[inf]
steinc1_100			10	71	0	**0.12**	1,460	3,217 (64.34%)	536.13	285	139,990	20,097	[inf]
steinc1_10			30	8	0	**0.13**	1,253	2,045 (40.90%)	161.54	1,502	29,800	5,843	[inf]
steinc1_100			30	71	0	**0.12**	3,104	2,531 (50.62%)	416.89	1,207	26,545	5,265	[inf]
steinc2_10			1	32	0	**0.11**	22	2,865 (57.30%)	10.75	112	92,822	21,332	[inf]
steinc2_100			1	328	0	**0.11**	22	2,865 (57.30%)	10.8	112	92,822	21,332	[inf]
steinc2_10			10	32	0	**0.13**	1,595	2,831 (56.62%)	308.48	189	68,055	9,182	[inf]
steinc2_100			10	328	0	**0.13**	1,244	3,125 (62.50%)	202.69	324	77,146	9,158	[inf]
steinc2_10			30	32	0	**0.13**	1,440	1,937 (38.74%)	105.54	2,549	30,787	10,650	[inf]
steinc2_100			30	328	0	**0.13**	1,365	1,937 (38.74%)	124.22	460	31,898	12,087	[66.77%]
steinc3_10			1	151	0	**0.12**	0	2,906 (58.12%)	4.99	29	166,681	19,320	[inf]
steinc3_100			1	1,519	0	**0.13**	0	2,906 (58.12%)	5.07	30	129,354	17,600	[inf]
steinc3_10			10	151	0	**0.13**	613	2,496 (49.92%)	66.16	266	105,634	19,968	[inf]
steinc3_100			10	1,519	0	**0.12**	307	2,356 (47.12%)	33.79	1,462	111,538	17,429	[inf]
steinc3_10			30	95	14	**0.2**	887	1,398 (27.96%)	48.15	1,810	79,964	13,165	[34.42%]
steinc3_100			30	968	45	**0.25**	1,719	1,591 (31.82%)	65.66	1,249	77,391	14,692	[91.68%]
steinc4_10			1	115	0	**0.11**	0	3,141 (62.82%)	2.99	127	102,621	20,354	[inf]
steinc4_100			1	1,148	0	**0.12**	0	3,147 (62.94%)	3.06	163	103,229	22,120	[inf]
steinc4_10			10	115	0	**0.12**	109	2,857 (57.14%)	12.08	291	77,732	16,095	[inf]
steinc4_100			10	1,148	0	**0.12**	182	2,931 (58.62%)	30.2	572	71,038	15,558	[inf]
steinc4_10			30	84	4	**0.14**	1,537	3,043 (60.86%)	102.04	1,589	53,224	14,747	[90.50%]
steinc4_100			30	854	8	**0.15**	1,785	1,816 (36.32%)	109.94	1,430	40,269	14,014	[96.39%]
steinc5_10			1	258	0	**0.12**	0	3,220 (64.40%)	1.31	92	759,835	28,310	[inf]
steinc5_100			1	2,600	0	**0.11**	0	3,226 (64.52%)	1.32	98	907,096	34,754	[inf]
steinc5_10			10	258	0	**0.12**	227	2,897 (57.94%)	14.48	901	141,186	16,085	[inf]
steinc5_100			10	2,600	0	**0.12**	78	2,893 (57.86%)	24.97	459	138,829	17,203	[inf]
steinc5_10			30	154	0	**0.12**	1,404	1,860 (37.20%)	138.3	1,196	74,855	15,840	[inf]
steinc5_100			30	1,584	0	**0.12**	683	1,619 (32.38%)	71.8	2,236	111,998	15,738	[14.36%]

in the GG-Hop formulation but this did not prove to be beneficial and this option was dropped after some preliminary tests.

3.3. *Summary of the Algorithmic Strategies and Branch-and-Cut Template*

We have tested the five algorithmic strategies summarized in Table 1. The general branch-and-cut template for these strategies is presented in Algorithm 2.

4. COMPUTATIONAL EXPERIMENTS

The branch-and-cut strategies just described were implemented within the CPLEX 9.1.3 framework with standard settings and run on an AMD Opteron machine with a 2,390 MHz CPU and 16 Gb RAM, under Linux. In all cases, branching priority was given to the variables associated with edges incident to the root vertex and the branching node selection was performed according to the best bound rule.

4.1. *Test Instances*

We have tested the proposed valid inequalities and branch-and-cut strategies on the sets of Steiner instances B and C obtained from the OR-Library [1]. We have adapted these instances to the STPRBH by using the terminal vertices as profitable vertices with revenue generated randomly according to a discrete uniform distribution over the interval $[1, 100]$. The Steiner vertices were attributed a zero

TABLE 8. Results - instances steinc, $h = 15$.

Instance					$S1$		$S3$			$S5$			
Name	n	$\|E\|$	b	Opt	B & B nodes	Seconds	B & B nodes	Cuts (%)	Seconds	B & B nodes	Sep	Hops	Seconds
steinc1_10	500	625	1	27	0	**0.48**	44	7,660 (43.77%)	43.99	100	1,293,143	25,266	[inf]
steinc1_100			1	274	0	**0.47**	44	7,660 (43.77%)	43.98	100	1,293,143	25,266	[inf]
steinc1_10			10	27	0	**0.47**	325	2,978 (17.02%)	70.68	2,823	57,394	8,468	[inf]
steinc1_100			10	274	0	**0.49**	797	5,114 (29.22%)	347.12	1,975	40,568	6,840	[11.67%]
steinc1_10			30	27	0	**0.49**	493	1,641 (9.38%)	49.8	5	6	16	1.74
steinc1_100			30	274	0	**0.48**	403	1,558 (8.90%)	29.79	5	6	16	1.74
steinc2_10			1	59	3	**2.55**	227	10,525 (60.14%)	520.89	117	1,127,462	16,528	[inf]
steinc2_100			1	604	3	**2.57**	227	10,525 (60.14%)	521.19	117	1,127,462	16,528	[inf]
steinc2_10			10	59	3	**2.57**	613	3,319 (18.97%)	79.3	2,423	34,708	5,454	[5.08%]
steinc2_100			10	604	3	**2.52**	1,840	5,177 (29.58%)	450.5	320	6,427	1,180	229.07
steinc2_10			30	53	216	58.32	12	290 (1.66%)	**2.57**	15	1,011	48	7.03
steinc2_100			30	546	319	224.28	211	1,114 (6.37%)	11.98	2	290	32	**3.17**
steinc3_10			1	439	3	**49.95**	77	10,263 (58.65%)	900.85	580,833	13,260	1,569	7,105.42
steinc3_100			1	4,463	3	**50.01**	65	10,126 (57.86%)	996.52	644,089	14,220	988	[inf]
steinc3_10			10	289	90	114.05	2,834	4,548 (25.99%)	397.77	82	683	120	**28.93**
steinc3_100			10	2,971	310	190.13	4,354	4,658 (26.62%)	663.3	464	1,447	192	**85.81**
steinc3_10			30	129	3	1.32	430	837 (4.78%)	24.25	0	41	8	**0.5**
steinc3_100			30	1,343	0	1	643	996 (5.69%)	35.2	0	56	8	**0.43**
steinc4_10			1	648	1	**19.28**	25	9,754 (55.74%)	368.48	126	1,395,060	23,054	[inf]
steinc4_100			1	6,566	1	**19.21**	25	9,672 (55.27%)	306.99	126	1,395,060	23,054	[inf]
steinc4_10			10	336	20	**15.05**	826	2,958 (16.90%)	163.98	5,722	36,147	5,306	2,740.75
steinc4_100			10	3,458	214	**77.28**	3,150	4,676 (26.72%)	509.52	1,152	97,257	12,709	[inf]
steinc4_10			30	134	13	**13.54**	460	1,424 (8.14%)	37.41	290	4,268	1,464	348.86
steinc4_100			30	1,380	31	**9.75**	119	1,021 (5.83%)	9.8	148	113	90	23.36
steinc5_10			1	1,248	0	**45.69**	121	11,855 (67.74%)	871.98	3,770	589,718	7,092	[inf]
steinc5_100			1	12,533	0	**45.58**	59	11,578 (66.16%)	760.04	3,064	529,800	7,034	[inf]
steinc5_10			10	494	8	**5.83**	1,693	3,817 (21.81%)	251.73	1,044	61,998	4,638	1,234.44
steinc5_100			10	5,032	22	**7.13**	1,400	3,540 (20.23%)	129.9	262	44,232	3,516	1,465.93
steinc5_10			30	182	11	**8.63**	1,349	2,558 (14.62%)	82.8	202	1,990	458	93.38
steinc5_100			30	1,857	93	22.24	638	1,668 (9.53%)	50.2	60	70	44	**9.97**

revenue. For each instance of the first group, 12 scenarios were analyzed, obtained from four values for the hop limit, $h = 3, 6, 9$, and 12, and three values for the budget, $b = s, s/5$, and $s/10$, where $s = \sum_{(i,j)\in E} c_{ij}$. For the second group of instances, we used $h = 5, 15$, and 25 and $b = s, s/10$, and $s/30$. Because vertex 1 is not always a terminal vertex in the original instances, the root vertex was chosen as the terminal vertex with the smallest index. The instance sizes range from 50 vertices and 63 arcs to 500 vertices and 625 arcs.

It is interesting to observe how the budget and hop constraints interact, making the solutions of different scenarios for the same instance completely different. Instance *Steinb*8 is a good example and is detailed in Table 2 which shows, for each scenario, the optimal solution value, the number of profitable vertices reached, and the percentage of the total revenue collected. The table also shows the budget limit and the sum of arc costs in the solution (Cost).

The table is linked to Figures 1 and 2 which depict the solutions. The values inside the vertices are the vertex numbers, whereas the values next to the arcs are their cost. A vertex is represented by a circle if its revenue is zero and by a square, otherwise.

For the first three scenarios, there is no budgetary limit and, therefore, the hop constraint is the only restriction preventing the solution from reaching all profitable vertices. For $h = 3$, only two vertices can be reached, yielding a total revenue of 85. For $h = 6$, the total revenue increases to 581 (given by seven profitable vertices) but it is only for $h = 9$ and $h = 12$ that all 19 profitable vertices are reached. A similar analysis can be made for a constant value of the hop limit, for example, $h = 9$: in this case, the reduction in the budget reduces the total collectable revenue from 836 ($b = s = 501$) to 761 ($b = s/5 = 100.2$) and finally to 537 ($b = s/10 = 50.1$). In the sequel, we describe the results for all the instances.

4.2. Complete Tests

We were first interested in the effect of adding violated inequalities (11) and (13) when solving the polynomial formulation GG-Hop with and without the relaxation of constraints (35). The tests showed that, although for some instances the cuts could help achieve an earlier convergence, it is not as a rule beneficial to add these constraints and, for this reason, strategies S2 and S4 were disregarded. Although constraints (11) and (13) have not been useful in our tests, the

TABLE 9. Results - instances steinc, $h = 25$.

Instance					$S1$		$S3$			$S5$			
Name	n	$\|E\|$	b	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds
steinc1_10	500	625	1	27	2	**8.8**	17	6,547 (21.82%)	14.49	28	18,727	982	222.51
steinc1_100			1	274	2	8.76	17	6,547 (21.82%)	14.55	28	18,727	982	222.87
steinc1_10			10	27	2	8.8	1,490	8,301 (27.67%)	878.27	28	1,655	272	43.76
steinc1_100			10	274	2	8.77	1,257	6,922 (23.07%)	682.47	28	1,655	272	43.81
steinc1_10			30	27	2	9.07	1,473	2,973 (9.91%)	230.37	7	0	38	**2.86**
steinc1_100			30	274	2	8.97	553	2,459 (8.20%)	56.54	7	0	38	**2.85**
steinc2_10			1	59	0	5.72	2	1,414 (4.71%)	**2.73**	0	178	82	7.24
steinc2_100			1	604	0	5.72	2	1,414 (4.71%)	**2.73**	0	178	82	7.3
steinc2_10			10	59	0	**5.64**	430	4,207 (14.02%)	57.57	0	178	82	7.26
steinc2_100			10	604	0	**5.69**	908	4,733 (15.78%)	183.89	0	178	82	7.26
steinc2_10			30	53	348	431.84	376	1,174 (3.91%)	24.98	17	1,001	60	**8.5**
steinc2_100			30	546	296	362.61	390	1,716 (5.72%)	24.9	14	292	30	**4.12**
steinc3_10			1	439	0	223.75	166	17,584 (58.61%)	257.3	0	1,002	8	**2.16**
steinc3_100			1	4,463	0	184.55	84	15,069 (50.23%)	120.87	0	1,002	8	**2.15**
steinc3_10			10	289	501	4,293.36	7,728	7,496 (24.99%)	3,309.14	64	35	56	**18.46**
steinc3_100			10	2,979	223	[inf]	8,601	8,351 (27.84%)	[0.23%]	204	1,001	64	**36.3**
steinc3_10			30	129	47	153.97	626	1,920 (6.40%)	44.77	0	0	8	**0.51**
steinc3_100			30	1,343	3	8.17	130	1,464 (4.88%)	13.47	0	6	8	**0.44**
steinc4_10			1	648	1	539.17	101	19,869 (66.23%)	[0.31%]	8	9,412	328	112.65
steinc4_100			1	6,566	1	542.52	66	19,077 (63.59%)	[inf]	8	9,412	328	112.65
steinc4_10			10	341	5	14.8	1,547	4,811 (16.04%)	141.98	3	0	4	**2.34**
steinc4_100			10	3,504	30	30.81	514	3,361 (11.20%)	70.68	0	0	4	**0.88**
steinc4_10			30	136	109	1,149.3	181	1,901 (6.34%)	26.07	5	0	10	**2.51**
steinc4_100			30	1,396	120	115.14	702	2,218 (7.39%)	57.45	46	0	18	**8.64**
steinc5_10			1	1,248	5	1,009.69	46	18,999 (63.33%)	150.38	3	54,477	224	**98.48**
steinc5_100			1	12,533	8	1,208.92	46	18,999 (63.33%)	150.31	3	54,477	224	**98.47**
steinc5_10			10	495	401	139.93	2,212	5,367 (17.89%)	550.42	49	1,088	54	**21.03**
steinc5_100			10	5,044	521	305.44	3,242	5,150 (17.17%)	878.85	21	545	36	**10.74**
steinc5_10			30	183	324	1,568.97	235	1,739 (5.80%)	18.7	81	0	40	**13.58**
steinc5_100			30	1,860	206	89.11	490	2,270 (7.57%)	29.82	76	0	16	**7.08**

development of theoretical comparisons between these and the GG-Hop model could prove an interesting research topic.

Strategies $S1$, $S3$, and $S5$ were tested on the two sets of instances described in Section 4.1. A maximum CPU time of 2 h was allowed for the solution of each scenario of each instance. In Tables 3–9 we present the detailed results of our computational tests. Besides the time needed by the algorithm, we also report the number of cuts (11) and (13) for strategy $S5$ and the number of cuts (35) for strategy $S3$. In this case, we indicate the percentage of the total number of constraints (35) that were added to the model. For each algorithm, the tables also show the number of branch-and-cut nodes explored. In these tables, when the time limit is not sufficient for the algorithm to converge to a proven optimal solution, we present the MIP gap in brackets. The entry "[inf]" indicates that no feasible solution was identified within the time limit.

For the scenarios with a small hop limit h, $S1$ is clearly superior. Indeed, it is the best for all instances *Msteinb* with $h = 3$, and in all instances *Steinc* with $h = 5$. The reason for this behavior seems to be the fact that formulation GG-Hop is extremely strong, and its size is reasonable for small values of h. This hypothesis can be confirmed by the fact that a great proportion of these instances are solved at the root node. For small values of h, $S3$ is also effective but less than $S1$. However, formulation GG-Hop quickly grows with h and, for this reason, the relaxation of constraints (35) eventually starts to pay off, as one can see in a direct comparison of algorithms $S1$ and $S3$ for the *Steinb* instances with $h = 12$. For instances *Steinc*, $S1$ seems to be always superior to $S3$.

Interestingly, Algorithm $S5$ behaves in a complete different manner. Indeed, for small values of h, the number of generated constraints (11) and (13) is huge, making the algorithm very inefficient. As soon as h grows, however, fewer cuts are needed for convergence and $S5$ clearly becomes the most efficient approach (see, for example, the results for instances *Steinb* for $h = 12$ in Table 6).

Table 10 summarizes the results of these tests. For the three possible solution methods and the two groups of instances, we indicate the number of scenarios for which the method was the best and the number of scenarios the method could not solve within the imposed time limit. One can observe again the clear polarity between strategies $S1$ and $S5$. Indeed, for instances with a small h, $S1$ seems to be the most efficient strategy, whereas for instances with a large h, $S5$ yields better results. As indicated earlier, this was somehow expected because for increasing values of h two phenomena happen: on the one hand, the size of formulation GG-Hop quickly increases and,

TABLE 10. Summary of results.

Group	h	$S1$ Best	$S1$ Unsolved	$S3$ Best	$S3$ Unsolved	$S5$ Best	$S5$ Unsolved
Steinb	3	54	0	1	0	0	29
Steinb	6	48	0	5	0	1	18
Steinb	9	21	0	8	0	25	3
Steinb	12	6	0	14	0	34	0
SteinC	5	30	0	0	0	0	30
SteinC	15	23	0	1	0	6	13
SteinC	25	3	1	2	3	20	0
Total		185	1	31	3	86	93

on the other hand, the number of cuts necessary to find a feasible solution in strategy $S5$ becomes smaller.

As a final remark, note that the instances used in the tests are rather sparse. There are two reasons for this: (a) several real network design problems occur on topologies defined by streets and neighborhoods which yield sparse graphs, and (b) hop constraints make more sense in sparse graphs, because in complete graphs there is always a direct connection between every pair of vertices. Nevertheless, we have performed some tests to better understand the effect of increasing density without changing the way of computing the budget parameter. Our results indicate that strategy $S1$ is importantly affected by the increase on density, becoming much slower. The performance of strategies $S3$ and $S5$, in turn, seems to be quite robust to arc density. These results are somehow expected since the increase of the number of arcs greatly affects the number of variables in $S1$, but has little influence in $S5$. The relaxation of constraints (35) becomes more evident as density increases, thus giving $S3$ a clear advantage over $S1$. We have performed further tests in which the density was multiplied by a factor θ and the budget values were taken as s, $s/5\theta$, and $s/10\theta$. In this case, $S3$ clearly becomes the best strategy.

5. CONCLUSIONS

We have proposed several formulations for a modified version of the Steiner Tree Problem with revenues, including budget and hop constraints. These formulations were used to develop branch-and-cut algorithms for the problem. Computational tests have shown that the proposed algorithms are capable of solving a majority of scenarios for instances with up to 500 vertices and 625 arcs. Moreover, our results indicate that the best algorithm depends on the number of allowed hops. For small hop values, algorithms based on position variables are clearly superior, whereas for large hop values the algorithm based on Dantzig-Fulkerson-Johnson formulation is the most efficient.

ACKNOWLEDGMENTS

The authors thank Stefan Ropke for kindly supplying his maxflow algorithm implementation and three anonymous referees for their valuable suggestions. The authors finally thank Luis Gouveia whose comments have helped improve both the content and the presentation of this article.

REFERENCES

[1] J.E. Beasley, OR-Library: Distributing test problems by electronic mail, J Oper Res Soc 41 (1990), 1069–1072.

[2] D. Bienstock, M.X. Goemans, D. Simchi-Levi, and D. Williamson, A note on the prize collecting traveling salesman problem, Math Program 59 (1993), 413–420.

[3] S.A. Canuto, M.G.C. Resende, and C.C. Ribeiro, Local search with perturbations for the prize-collecting Steiner tree problem in graphs, Networks 38 (2001), 50–58.

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

[5] S. Chopra and M.R. Rao, The Steiner tree problem. II. Properties and classes of facets, Math Program 64 (1994), 231–246.

[6] R. Cole, R. Hariharan, M. Lewenstein, and E. Porat, A faster implementation of the Goemans-Williamson clustering algorithm, SODA'01: Proc Twelfth Annual ACM-SIAM Sympo Discrete Algorithms, Washington, 2001, pp. 17–25.

[7] A.M. Costa, J.-F. Cordeau, and G. Laporte, Steiner tree problems with profits, INFOR 44 (2006), 99–115.

[8] R. Courant and H. Robbins, What is mathematics? Oxford University Press, New York, 1941.

[9] A.S. Cunha, A. Lucena, N. Maculan, and M.G.C. Resende, A relax and cut algorithm for the prize collecting Steiner problem in graphs, Proc Math Program Rio, Búzios, Brazil, 2003, pp. 72–78.

[10] G. Dahl, L. Gouveia, and C. Requejo, "On formulations and methods for the hop-constrained minimum spanning tree problem," Handbooks of telecommunications, P. Pardalos and M.G.C. Resende (Editors), Springer, New York, 2006, pp. 493–515.

[11] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson, Solution of a large-scale traveling-salesman problem, Oper Res 2 (1954), 393–410.

[12] M. Desrochers and G. Laporte, Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints, Oper Res Lett 10 (1991), 27–36.

[13] C. Feremans, M. Labbé, and G. Laporte, A comparative analysis of several formulations for the generalized minimum spanning tree problem, Networks 39 (2002), 29–34.

[14] M. Fischetti, Facets of two Steiner arborescence polyhedra, Math Program 51 (1991), 401–419.

[15] M. Garcia, Árvores com restrições de diâmetro, Master thesis, University of Lisbon, 1994 (in Portuguese).

[16] M.R. Garey, R.L. Graham, and D.S. Johnson, The complexity of computing Steiner minimal trees, SIAM J Appl Math 32 (1977), 835–859.

[17] E.N. Gilbert and H.O. Pollak, Steiner minimal trees, SIAM J Appl Math 16 (1968), 1–29.

[18] M.X. Goemans, The Steiner tree polytope and related polyhedra, Math Program 63 (1994), 157–182.

[19] M.X. Goemans and D.P. Williamson, A general approximation technique for constrained forest problems, SIAM J Comput 24 (1995), 296–317.

[20] B.L. Golden, S. Raghavan, and D. Stanojević, The prize-collecting generalized minimum spanning tree problem, J Heuristics 14 (2008), 69–93.

[21] L. Gouveia, Using the Miller-Tucker-Zumlin constraints to formulate a minimal spanning tree problem with hop constraints, Comput Oper Res 22 (1995), 959–970.

[22] L. Gouveia, Multicommodity flow models for spanning trees with hop constraints, Eur J Oper Res 95 (1996), 178–190.

[23] L. Gouveia, Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints, INFORMS J Comput 10 (1998), 180–188.

[24] L. Gouveia, "Using hop-indexed models for constrained spanning and Steiner tree models," Telecommunications network planning, B. Sansò and P. Soriano (Editors), Kluwer, Boston, 1999, pp. 21–32.

[25] L. Gouveia and C. Requejo, A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem, Eur J Oper Res 132 (2001), 539–552.

[26] F.K. Hwang, D.S. Richards, and P. Winter, The Steiner tree problem, North-Holland, Amsterdam, 1992.

[27] D.S. Johnson, M. Minkoff, and S. Phillips, The prize collecting Steiner tree problem: theory and practice, Proc 11th ACM-SIAM Sympo Discrete Algorithms, San Francisco, 2000, pp. 760–769.

[28] B.N. Khoury, P.M. Pardalos, and D.-Z. Du, A test problem generator for the Steiner tree problem in graphs, ACM Trans Math Software 19 (1993), 509–522.

[29] G.W. Klau, I. Ljubić, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, and R. Weiskircher, Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem, Springer Lecture Notes Comput Sci 3102 (2004), 1304–1315.

[30] T. Koch and A. Martin, Solving Steiner tree problems in graphs to optimality, Networks 32 (1998), 207–232.

[31] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti, Solving the prize-collecting Steiner tree problem to optimality, Proc ALENEX, 7th Workshop Algorithm Engineering Experiments, Vancouver, 2005.

[32] A. Lucena, Tight bounds for the Steiner problem in graphs, Paper Presented at TIMS XXX - SOBRAPO XXIII Joint international meeting, Rio de Janeiro, 1991.

[33] A. Lucena and M.G.C. Resende, Strong lower bounds for the prize collecting Steiner problem in graphs, Discrete Appl Math 141 (2004), 277–294.

[34] T. Magnanti and S. Raghavan, Strong formulations for network design problems with connectivity requirements, Networks 45 (2005), 61–79.

[35] F. Margot, A. Prodon, and T.M. Liebling, Tree polyhedron on 2-tree, Math Program 63 (1994), 183–192.

[36] C.E. Miller, A.W. Tucker, and R.A. Zemlin, Integer programming formulations and traveling salesman problems, J Assoc Comput Mach 7 (1960), 326–329.

[37] R. Prim, Shortest connection networks and some generalizations, Bell Syst Technol J 36 (1957), 1389–1401.

[38] A. Segev, The node-weighed Steiner tree problem, Networks 17 (1987), 1–17.

[39] A.W. Shogan, Constructing a minimal cost spanning tree subject to resource constrains and flow requirements, Networks 13 (1983), 169–190.

[40] S. Voß, The Steiner tree problem with hop constraints, Annals Oper Res 86 (1999), 321–345.