ORIGINAL ARTICLE

A comparison of node-based and arc-based hop-indexed formulations for the Steiner tree problem with hop constraints

Bernard Fortz[1] | Luis Gouveia[2] | Pedro Moura[3]

[1]Computer Science Department, Université libre de Bruxelles (ULB), Bruxelles, Belgium, and INOCS, INRIA Lille Nord-Europe, France

[2,3]Departamento de Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa and CMAFcIO - Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Lisboa, Portugal

Correspondence
Luis Gouveia
Email: legouveia@fc.ul.pt

Funding information
Supported by National Funding from FCT - Fundação para a Ciência e a Tecnologia, under the project: UIDB/04561/2020

We study the relation between the linear programming relaxation of two classes of models for the Steiner tree problem with hop constraints. One class is characterized by having hop-indexed arc variables. Although such models have proved to have a very strong linear programming bound, they are not easy to use because of the huge number of variables. This has motivated some studies with models involving fewer variables that use, instead of the hop-indexed arc variables, hop-indexed node variables.

In this paper we contextualize the linear programming relaxation of these node-based models in terms of the linear programming relaxation of known arc-based models. We show that the linear programming relaxation of a general node-based model is implied by the linear programming relaxation of a straightforward arc-based model.

KEYWORDS
Steiner tree, hop constraint, network design, linear programming relaxation, integer programming, hop-indexed model

Consider a graph $G = (N_0, A)$, with node set $N_0 = \{0, 1, \ldots, n\}$ and arc set A, with a nonnegative cost c_a associated to each arc $a \in A$, an integer hop limit H, and a set of required terminal nodes $R \subseteq N_0$ with $0 \in R$. The hop-constrained Steiner tree problem (HSTP) consists of determining an arborescence $G' = (N', T)$ rooted at 0, spanning a subset $N' \supseteq R$, so that the unique path from 0 to each terminal $r \in R$ contains at most H arcs, and the cost of selected arcs $c(T) := \sum_{a \in T} c_a$ is minimized. This problem was first introduced by Voß [16]. Besides its natural application in telecommunications, studies on this problem and related ones have been of interest because they include state of the art research on so-called hop-indexed and/or layered graph formulations which are topical for problems with such distance-like constraints, see, e.g., [1, 2, 4, 5, 9, 11, 12, 13, 14].

The hop-indexed models described in these works are characterized by having hop-indexed arc variables z_{ij}^h indicating whether arc (i, j) is in position h (that is, the single path from node 0 to node i contains $h-1$ arcs) in the solution. Although such models have proved to have a very strong linear programming bound, they are not easy to use because of the huge number of variables. This has motivated some studies with models involving fewer variables and that use node variables v_i^h indicating whether node i is in position h, instead of the arc variables z_{ij}^h. Recently, Sinnl and Ljubić [15] have presented one such model for the budget constrained hop constrained Steiner tree problem, first introduced by Costa *et al.* [3], where the objective is the maximization of the revenue.

In this paper we want to contextualize the linear programming relaxation of the node-based model in terms of the linear programming relaxation of known arc-based models. The arguments given next suggest that, in general, a node-based model has a weak linear programming bound, at least when compared with an arc-based model. First, observe that an arc variable provides more information than a node variable does; e.g., the node variable v_i^h indicates whether node i is in position h and the arc variable z_{ji}^h indicates whether node i is in position h AND the arc entering node i is coming from node j. Thus, defining a model with arc variables should be easier (or stronger in terms of the linear programming relaxation bound) than writing a valid model with node variables. We can make this argument more formal with equalities such as $v_i^h = \sum_{(k,i)\in A} z_{ki}^h$, relating the two sets of variables. If we add such equalities to an arc-based model, in theory we could project out the arc variables and obtain a model defined only on the node variables with an equivalent LP relaxation. In several cases, it may not be easy to find the whole set of projected inequalities, however we can obtain a subset of inequalities that still result in a valid model (although with a weaker linear programming bound). In fact, this is what happens with the pair of models, $gBNH$ and AH, discussed later in the paper. The dominance of an arc-based model over a node-based model is a general observation: the linking equalities allow any node-based model to be rewritten as an arc model simply by using them to replace the v_i^h variables by z_{ij}^h variables and thus by simple substitution the arc model is always as strong as the node model.

In this paper we show that the linear programming relaxation of the node-based model (including a large set of generalized inequalities) is implied by the linear programming relaxation of a “simple" arc-based model that was presented formerly by Gouveia [8] for the Spanning Tree Problem and easily adapted for the more general problem studied in this paper. We have used the term “simple", because the inequalities defining this model are a weaker version of a rather small subset of a more general class of inequalities, the so-called layered graph cuts that are included in the model proposed in by Gouveia *et al.* [11]. This model is, as a far we know, the strongest model known for this problem.

To simplify the notation, and before presenting the formulations, we define the following sets: $H_1 := \{1, \ldots, H\}$, $H_2 := \{2, \ldots, H\}$, $N_1 := N_0 \setminus \{0\} = \{1, \ldots, n\}$, $R_1 := R \setminus \{0\}$.

1 | NODE-BASED HOP-INDEXED MODEL

In this section we discuss node-based hop-indexed models for the HSTP. We classify these models either as “forward" models or “backward" models: a “forward" model is characterized by constraints forcing a node to be at distance h if there is an arc entering the node coming from a node at distance $h-1$; a “backward" model is characterized by constraints indicating that a node must be at distance $h-1$ if there exists an arc leaving that node to a node at distance h. We observe that although the more general models of the two classes are equivalent, there are a few relevant differences in the two modelling views. For this reason, we have divided this section into two subsections, dedicated to each one of the two classes.

1.1 | Forward models

Most of the material in this section is adapted from Sinnl and Ljubić [15] where the authors proposed several node-based models for the Steiner tree problem with revenues, budget and hop-constraints. The model that we adapt here for the HSTP uses binary variables y_i to indicate if node $i \in N_1$ belongs to N', binary variables x_{ij} to indicate if arc $(i, j) \in A$ belongs to T, and binary hop-indexed node variables v_i^h to indicate if node $i \in N_1$ is at distance $h \in H_1$ from root node 0 in G'. Consider also, the following set of constraints that all feasible solutions must satisfy

$$\sum_{(i,j)\in A} x_{ij} = y_j \quad j \in N_1, \tag{1}$$

$$\sum_{h\in H_1} v_i^h = y_i \quad i \in N_1, \tag{2}$$

$$v_j^1 = x_{0j} \quad j : (0, j) \in A, \tag{3}$$

$$v_i^{h-1} + x_{ij} \le v_j^h + 1 \quad (i, j) \in A,\ i \in N_1,\ h \in H_2, \tag{4}$$

$$v_i^H + x_{ij} \le 1 \quad (i, j) \in A,\ i \in N_1, \tag{5}$$

$$y_i = 1 \quad i \in R_1, \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in A, \tag{7}$$

$$y_i \in \{0, 1\} \quad i \in N_1, \tag{8}$$

$$v_i^h \in \{0, 1\} \quad i \in N_1,\ h \in H_1 \tag{9}$$

Constraints (1) impose that each node, except the root node 0, has exactly one entering arc if it belongs to the set of nodes selected in the solution, or zero otherwise. Constraints (2) state that any node, other than the root, belonging to the solution is at a distance between 1 and H from the root, while constraints (3) impose that a node connected directly to the root is at distance 1 from it. Constraints (4) state that if a node i is at distance $h - 1$ from the root, and arc (i, j) belongs to the solution, then node j is at distance h from the root. Similarly, constraints (5) forbid an arc leaving a node which is at the maximum distance H from the root. Nodes in R_1 are forced to belong to the solution by constraints (6) (node 0 implicitly belongs to the solution) and constraints (7), (8) and (9) ensure that all variables are binary.

As pointed out by Sinnl and Ljubić [15], the formulation (1)-(9) is not sufficient to get a valid formulation for HSTP as connectivity between the root and the required terminal nodes is not ensured, as illustrated in the example in Figure 1. The example consists of a complete graph with 4 nodes ($n = 3$ plus the root node) with $H = 3$ and required nodes, $R = \{0, 1, 3\}$. The arcs in the example correspond to the arc variables $x_{01} = x_{23} = 1$. The remaining variable values are $y_1 = v_1^1 = 1$, $y_3 = v_3^2 = 1$ and zero for all other variables. This "solution" satisfies all constraints (1)-(9) but is obviously not feasible as there is no path from the root node to node 3.

To enforce connectivity, one can still follow Sinnl and Ljubić [15] and add a set of the well-known generalized cut constraints, or alternatively and as also pointed in their paper, we can add a smaller subset of generalized subtour elimination constraints of size two:

$$x_{ij} + x_{ji} \le y_i \quad i, j \in N_1 \tag{10}$$

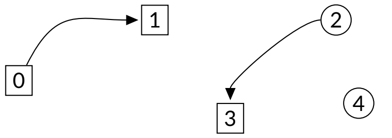

FIGURE 1 Infeasible solution for the HSTP.

We refer to Theorem 1 in Sinnl and Ljubić [15] for a proof that this compact model is valid for the Steiner tree problem with revenues, budget and hop-constraint, and which also applies for the Steiner version studied in this paper.

In fact, to ensure connectivity between the root node and any required node and to guarantee the validity of the forward model (1)-(9), we only need to consider the following “weaker" version of constraints (10):

$$x_{ij} \leq y_i \quad i, j \in N_1 \tag{11}$$

Observe that in the example in Figure 1, this constraint is not satisfied for $(i, j) = (2, 3)$ and $i = 2$. We show next that adding these constraints to the formulation (1)-(9) guarantees the connectivity of any solution to the HSTP.

Proposition 1 *Formulation (1), (2), (3), (4), (5), (6), (7), (8), (9) and (11) is a valid formulation for the HSTP.*

Proof We first observe that due to constraints (2) and (4), a solution to this model cannot contain circuits. To see this, consider a circuit $C = \{i_1, i_2, \ldots, i_k\}$. Using constraints (4) in a circular fashion, starting from node i_1 for example and moving forward, we would obtain $v_{i_1}^h = v_{i_1}^{h+k} = 1$, for a given value of h. But this is in contradiction with constraint (2) for node i_1.

We show next that for any node j such that $j \in R_1$, there exists a path $P = \{j_1, j_2, \ldots, j_k\}$ such that $j_1 = 0$ and $j_k = j$. We have $y_j = y_{j_k} = 1$ since $j \in R_1$. Constraints (1) guarantee that there must exist one and only one arc entering node j_k, say arc (j_{k-1}, j_k). If $j_{k-1} = 0$ we have found the required path. Otherwise, we have that $y_{j_{k-1}} = 1$ either because $j_{k-1} \in R_1$ or because of the new constraints (11). Repeating the process we find a node j_{k-2}, such that arc $(j_{k-2}, j_{k-1}) \in A$ and either j_{k-2} is the root node or $y_{j_{k-2}} = 1$. By repeating this process and since V is finite and the solution cannot contain cycles we obtain a node $y_1 = 0$ giving the required path.

Observe that this reasoning also applies to a node $j \notin R_1$ such that $y_j = 1$ and for which there is no arc emanating from it. However, a solution containing such a node would not be optimal. Thus, a solution to this model must contain a single path from the root node to any node in the solution. □

A strengthening of (4) and a generalization of the resulting strengthened inequality is also presented by Sinnl and Ljubić [15]. The strengthening is as follows:

$$v_i^H + v_i^{h-1} + x_{ij} \leq y_i + v_j^h \quad i \in N_1,\ (i, j) \in A, h \in H_2 \tag{12}$$

Observe that constraints (12) are obtained from (4) by replacing 1 by y_i on the right-hand side of (4) and adding v_i^H on the left-hand side. One question is to know whether constraints (12), in place of (4) in model (1)-(9), are sufficient to define a valid model or if, as before, we need to add extra constraints such as (11) or (10) to guarantee that any required node is connected to the root. We will argue next, that from an integer point of view, if $H \geq 3$, the new constraints (12) imply constraints (11), that is, if $x_{ij} = 1$ then $y_i = 1$. Let us assume that for a given arc (i, j) and a position h, we have $x_{ij} = 1$, $y_i = 0$; then we must have $v_j^h = 1$ for every $h \geq 2$ for the inequalities (12) (there are at least two) to hold. But this is inconsistent with constraint (2), since there can only be at most one $v_j^h = 1$. This argument leads to the next result.

Proposition 2 *Formulation (1), (2), (3), (12), (5), (6), (7), (8), (9) is a valid formulation for the HSTP when* $H \geq 3$.

Note that the disconnected solution in Figure 1 shows that this result is not valid for $H = 2$.
The generalization of constraints (12) is obtained by considering sets of variables associated to different distance values, *e.g.*, a specific subset of distance values, $S \subseteq H_2$, leading to:

$$v_i^H + \sum_{h \in S} v_i^{h-1} + x_{ij} \leq y_i + \sum_{h \in S} v_j^h \quad i \in N_1,\ (i,j) \in A, S \subseteq H_2 \tag{13}$$

Observe that, when $|S| = 1$ we obtain the original inequalities (12). Also, by eliminating y variables with the help of constraints (2), we obtain

$$x_{ij} \leq \sum_{h \in H_2 \setminus S} v_i^{h-1} + \sum_{h \in S} v_j^h \quad i \in N_1,\ (i,j) \in A,\ S \subseteq H_2 \tag{14}$$

which is the form presented by Sinnl and Ljubić [15]. Although exponential in number, these inequalities can be separated in polynomial time.

1.2 | Backward models

Using the same set of variables, a simple version of backward inequalities would be the symmetric version of (4)

$$v_j^h + x_{ij} \leq v_i^{h-1} + 1 \quad i \in N_1, (i,j) \in A,\ h \in H_2 \tag{15}$$

As shown before, formulation (1)-(9) does not ensure connectivity between the root node and the required terminal nodes. In contrast, and as proven in the following proposition, the backwards modelling approach, that is replacing constraints (4) by (15), leads to a valid formulation without the need to add constraints such as (10) or (11):

Proposition 3 *Formulation (1), (2), (3), (15), (5), (6), (7), (8), (9) is a valid formulation for the HSTP.*

Proof As noted before, constraints (2) and (15), guarantee that a solution to this model cannot contain circuits. The reasoning is similar to the one in the proof of Proposition 1, but this time in a backward way. Now, consider a node j in R_1 and let (i,j) be the corresponding arc entering this node, that is, $y_j = 1$ and $x_{ij} = 1$. Also, due to constraint (2) for node j, there exists a hop index $h^* \in H_1$ such that $v_j^{h^*} = 1$.
Assume $i \neq 0$ (if $i = 0$, we have a path from the root to node j). Then constraint (15) for arc (i,j) and $h = h^*$ becomes $1 \leq v_i^{h^*-1}$ and thus we have $v_i^{h^*-1} = 1$. Constraints (2) imply that $y_i = 1$, and constraints (1) guarantee that there exists an arc entering node i. By repeating the reasoning above, we conclude that a solution to this model must contain a path from the root node to any node. □

Similarly to the strengthening of (4) presented in the previous section, constraints (15) can also be strengthened to take into account border effects at distance 1 and as well as including the fact that some nodes may not be included in the solution. This leads to

$$v_j^1 + v_j^h + x_{ij} \leq y_j + v_i^{h-1} \quad i \in N_1, (i,j) \in A,\ h \in H_2 \tag{16}$$

Finally, in the same way that (12) are generalized into (13), the inequalities (16) can be generalized into

$$v_j^1 + \sum_{h \in S'} v_j^h + x_{ij} \leq y_j + \sum_{h \in S'} v_i^{h-1} \quad i \in N_1,\ (i,j) \in A, S' \subseteq H_2 \tag{17}$$

As the following proposition shows, these constraints can also be shown to be equivalent to (14) by using constraints (2).

Proposition 4 *In the presence of constraints (2), constraints (17) are equivalent to constraints (14).*

Proof Consider constraint (17) for a given node $i \in N_1$, an arc $(i,j) \in A$ and a subset $S' \subseteq H_2$. After replacing y_j in (17) by the left-hand side of equality (2) for node j and cancelling equal terms we obtain constraint (14) for a subset $S = H_2 \setminus S'$. □

Thus, we have proved that the three sets of constraints (13), (14) and (17) are equivalent. This proposition also gives an indirect proof that the generalized backward inequalities (17) are valid.

Table 1 provides a general view of the main constraints from the two classes of models, in particular, the linking constraints between the node variables y_i and the node-hop variables v_i^h in each model. Constraints (14) stand as the bridge between the two generalized strong models. That is, constraints (13) that characterize the forward generalized strong model are shown to be equivalent to constraints (17) of the the backward generalized strong model via the intermediate constraints (14).

constraints		Forward Models		Backward Models
common		(1), (2), (3), (6), (7), (8), (9)		
linking	weak	(4)[1]		(15)
	strong	(12)		(16)
	generalized strong	(13)	(14)	(17)

[1] needs extra constraints such as (10) or (11) for validity

TABLE 1 Valid hop-indexed node models for the HSTP: Forward and Backward

In Section 3 we show that the generalized constraints (17) (alternatively, (13) and (14)) are implied by a compact hop-indexed arc-based model.

2 | ARC-BASED HOP-INDEXED MODEL

The arc-based model presented in this section was first described by Gouveia [8] for the minimum spanning tree problem with hop constraints. The main idea of the model, more precisely constraints (21) and (22) to be described next, is to show that hop-indexed arc variables can easily be used to guarantee the hop limit as well as the connectivity of the solution (by using a backwards chain reasoning from any arc to the root node). This “arc-based hop-indexed" model is easily adapted for the HSTP. It uses variables y_i and x_{ij} as in the previous model and in addition, uses binary variables z_{ij}^h to indicate if arc (i,j) is in position h in the path from 0 to j.

$$\sum_{(i,j)\in A} x_{ij} = y_j \quad j \in N_1 \tag{18}$$

$$z^1_{0j} = x_{0j} \quad j : (0,j) \in A \tag{19}$$

$$\sum_{h\in H_2} z^h_{ij} = x_{ij} \quad i \in N_1,\ (i,j) \in A \tag{20}$$

$$\sum_{(k,i)\in A, k\neq 0} z^{h-1}_{ki} \geq z^h_{ij} \quad i \in N_1,\ (i,j) \in A,\ h \in H_2,\ h \geq 3 \tag{21}$$

$$z^1_{0i} \geq z^2_{ij} \quad i,j \in N_1 : (0,i),(i,j) \in A \tag{22}$$

$$y_i = 1 \quad i \in R_1, \tag{23}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A \tag{24}$$

$$y_i \in \{0,1\} \quad i \in N_1 \tag{25}$$

$$z^h_{ij} \in \{0,1\} \quad i \in N_1,\ (i,j) \in A,\ h \in H_2 \tag{26}$$

$$z^1_{0j} \in \{0,1\} \quad (0,j) \in A \tag{27}$$

Apart from constraints, (18), (23)-(25) that are the same as in the previous model, constraints (19) and (20) link the hop-indexed arc variables z^h_{ij} with the arc variables x_{ij}. Observe that each arc $(0,j) \in A$ can only be in position 1 in the solution and this is the reason why variables z^h_{0j} are defined only for $h = 1$. The remaining arcs in A can be in any position from 2 to H, therefore, variables z^h_{ij} for $(i,j) \in A$, are defined for $h \in H_2$. Constraints (21) guarantee that, if arc (i,j) leaves node $i \in N_1$ at position $h \geq 3$, then one arc $(k,i) \neq (0,i)$ enters that same node i at position $h-1$. Furthermore, since 2-cycles are not allowed, we can strengthen inequalities (21) by stating that $(k,i) \neq (j,i)$

$$\sum_{(k,i)\in A, k\neq 0,j} z^{h-1}_{ki} \geq z^h_{ij} \quad i \in N_1,\ (i,j) \in A,\ h \in H_2,\ h \geq 3 \tag{28}$$

Constraints (22) correspond to constraints (21) written for nodes directly connected to the root node. We denote by AH the model defined by constraints (18)-(27) and by sAH the model equivalent to AH with constraints (21) replaced by the stronger version, (28).

Constraints (19) and (20) can also be used to the remove variables x_{ij} from the formulation, thus obtaining a model with fewer variables. These constraints are not needed to provide a valid formulation for the problem. They are included here in order to establish the relation proved in the next section.

3 | RELATIONS BETWEEN THE FORMULATIONS

In this section we compare the linear programming relaxation of the models presented in the previous sections, namely, the AH model defined by constraints (18) - (27) and the *generalized Backward Node-based Hop-indexed* ($gBNH$) model presented in Section 1, defined by constraints (1), (2), (3), (17), (5), (6), (7), (8), (9).

Let $Model_L$ be the linear programming relaxation of a given $Model$ and $Feas(Model_L)$ its set of feasible solutions. Also, for a given polyhedron $Q \subseteq \mathbb{R}^{n\times m}$, the projection of Q in the subspace $\mathbb{R}^n$ is defined as $proj_{(x)}Q = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m \text{ such that } (x,y) \in Q\}$.

The following proposition, which is the main result of the paper, relates the AH_L model with the $gBNH_L$ model.

For that purpose, we add the following linear equalities to model AH, defining the v_j^h variables in terms of the z_{ij}^h variables

$$v_i^h = \sum_{(k,i)\in A, k\neq 0} z_{ki}^h \quad i \in N_1,\, h \in H_2 \tag{29}$$

$$v_i^1 = z_{0i}^1 \quad i \in N_1 \tag{30}$$

and the v_i^h domain constraints (9). We denote by $AH+$ the model AH augmented with these equalities. Observe that (29) and (30) are only definitional and adding them to model AH does not change the LP value. Model $AH+$ was created to formalize a relation between the two models, AH_L and $gBNH_L$.

Proposition 5 *The projection of $Feas(AH+_L)$ onto the variable space of $gBNH$ is contained in $Feas(gBNH_L)$,*

$$proj_{(x,y,v)}\,(Feas(AH+_L)) \subseteq Feas(gBNH_L)$$

Moreover, this inclusion can be strict.

Proof We will show next that the constraints of model $gBNH_L$ are implied by the constraints of model $AH+_L$, namely constraints (2), (3), (17) and (5) (the remaining constraints are straightforwardly satisfied).

- Constraints (3) are implied by constraints (19) and (30).
- For a given node $i \in N_1$, adding constraints (29) for all $h \in H_2$ together with (30) results in

  $$\sum_{h\in H_2} v_i^h + v_i^1 = \sum_{h\in H_2} \sum_{(k,i)\in A, k\neq 0} z_{ki}^h + z_{0i}^1$$

  On the other hand, adding (20) for all $(i,j) \in A, i \in N_1$ together with (19) and using (18) results in

  $$\sum_{(i,j)\in A, i\neq 0} \sum_{h\in H_2} z_{ij}^h + z_{0j}^1 = \sum_{(i,j)\in A, i\neq 0} x_{ij} + x_{0j} = y_j$$

  Thus, constraints (2) are also satisfied for every $i \in N_1$.
- For a given node $i \in N_1$ and an arc (i,j), constraints (29) together with (21) and constraints (30) together with (22) imply that $z_{ij}^h \leq v_i^{h-1}$, $h \in H_2$. Adding $\sum_{(k,j)\in A, k\neq 0, i} z_{kj}^h$ to both sides of these inequalities and using (29) we obtain

  $$v_j^h \leq v_i^{h-1} + \sum_{(k,j)\in A, k\neq 0, i} z_{kj}^h$$

  For a given set $S \subseteq H_2$, adding the previous inequalities for $h \in S$ we obtain

  $$\sum_{h\in S} v_j^h \leq \sum_{h\in S} v_i^{h-1} + \sum_{(k,j)\in A, k\neq 0, i} \sum_{h\in S} z_{kj}^h$$

  From constraints (20), we have $\sum_{h\in S} z_{kj}^h \leq x_{kj}$, $k \in N_1$, $(k,j) \in A$, therefore the right-hand side of the previous inequality can be lifted up (using constraints (18), (19) and (30))

$$\sum_{h\in S} v_j^h \le \sum_{h\in S} v_i^{h-1} + \sum_{(k,j)\in A, k\neq 0,i} x_{kj} = \sum_{h\in S} v_i^{h-1} + y_j - x_{ij} - x_{0j} = \sum_{h\in S} v_i^{h-1} + y_j - x_{ij} - z_{0j}^1 = \sum_{h\in S} v_i^{h-1} + y_j - x_{ij} - v_j^1$$

therefore constraint (17) is also satisfied for any $i \in N_1$ and $S \subseteq H_2$.

- For a given node $i \in N_1$ and arc $(i,j) \in A$, adding (21) for all $h \in H_2, h \ge 3$ together with (22) and using (20) yelds

$$\sum_{h\in H_2, h\ge 3} \sum_{(k,i)\in A, k\neq 0} z_{ki}^{h-1} + z_{0i}^1 \ge \sum_{h\in H_2, h\ge 3} z_{ij}^h + z_{ij}^2 = x_{ij}$$

Note that the left-hand side can be rearranged and using the previous proof that constraints (2) are satisfied,

$$\sum_{h\in H_2, h\ge 3} \sum_{(k,i)\in A, k\neq 0} z_{ki}^{h-1} + z_{0i}^1 = \sum_{(k,i)\in A, k\neq 0} \sum_{h\in H_2} z_{ki}^h - \sum_{(k,i)\in A, k\neq 0} z_{ki}^H + z_{0i}^1 = y_i - \sum_{(k,i)\in A, k\neq 0} z_{ki}^H$$

Therefore, using equality (29) for node i and $h = H$ we obtain $x_{ij} \le y_i - v_i^H \le 1 - v_i^H$ and thus constraint (5) is satisfied and we conclude the proof of inclusion. □

To show that this inclusion can be strict, consider an example (see Figure 2) consisting of a complete graph with four nodes and the root node, with required nodes 2 and 4, and $H = 3$. A feasible solution to $gBNH_L$ may be represented by the subgraph where the values close to the arcs represent the x_{ij} values, the values close to the required nodes represent the v_i^h values for $h = 1, 2, 3$, respectively, and $y_2 = y_4 = 1$. All other variables have a zero value.

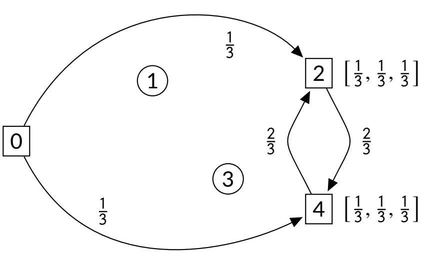

FIGURE 2 Feasible solution to $gBNH_L$.

No feasible solution to $AH+_L$ can be obtained from this solution (x, y, v) from $Feas(gBNH_L)$. In fact, from constraints (19) and (20) we have $z_{02}^1 = \frac{1}{3}$, $z_{04}^1 = \frac{1}{3}$, $z_{24}^2 + z_{24}^3 = \frac{2}{3}$ and $z_{42}^2 + z_{42}^3 = \frac{2}{3}$.
On the other hand, from constraints (21) we have $z_{24}^3, z_{42}^3 \le 0$, therefore $z_{24}^2 = z_{42}^2 = \frac{2}{3}$ which violates constraints (22) for the root arcs

$$\frac{1}{3} = z_{02}^1 \ge z_{24}^2 = \frac{2}{3} \quad \text{and} \quad \frac{1}{3} = z_{04}^1 \ge z_{42}^2 = \frac{2}{3}$$

□

One relevant observation is that this result shows that we obtained strict dominance over the node model $gBNH$ using the weaker arc model AH. Thus, one other question is to know what can be obtained by using the stronger model sAH defined by constraints (18) - (20), (28), (22) - (27). The next result gives a partial answer to this question by showing that the generalized subtour elimination constraints of size two (10) are implied by the stronger model sAH.

Proposition 6 *The generalized subtour elimination constraints of size two (10) are redundant if included in the sAH model.*

Proof For a given arc (i,j), we start by adding the term z_{ji}^{h-1} to both sides of the inequalities (28), the "missing" term on the summation corresponding to arc (j,i), leading to the following constraints that might be viewed as a kind of hop-indexed subtour elimination constraint of size 2,

$$\sum_{(k,i)\in A, k\neq 0} z_{ki}^{h-1} \geq z_{ij}^{h} + z_{ji}^{h-1}, \quad h \in H_2, h \geq 3$$

Next, we add these inequalities for all $h \in H_2, h \geq 3$ together with constraint (22) for arc (i, j) leading to,

$$z_{0i}^{1} + \sum_{(k,i)\in A, k\neq 0} \sum_{h\in H_2, h\geq 3} z_{ki}^{h-1} \geq z_{ij}^{2} + \sum_{h\in H_2, h\geq 3} z_{ij}^{h} + \sum_{h\in H_2, h\geq 3} z_{ji}^{h-1}$$

Using the equality constraints (19) and (20) on the above inequality we obtain

$$x_{0i} + \sum_{(k,i)\in A, k\neq 0} \left(x_{ki} - z_{ki}^{H}\right) \geq x_{ij} + \left(x_{ji} - z_{ji}^{H}\right)$$

Finally, using the indegree constraint (18) we obtain constraint (10) for the set $S = \{i, j\}$,

$$y_i \geq x_{ij} + x_{ji} - z_{ji}^{H} + \sum_{(k,i)\in A, k\neq 0} z_{ki}^{H} = x_{ij} + x_{ji} + \sum_{(k,i)\in A, k\neq 0, j} z_{ki}^{H} \geq x_{ij} + x_{ji}$$

□

Other inequalities of interest can probably be derived from the stronger model sAH. This is an open question that we leave for the future work.

4 | COMPUTATIONAL RESULTS

In this section, we compare some of the models introduced in the previous sections in terms of the Linear Programming (LP) relaxation bounds and CPU times to obtain the optimal integer solution. The tests were performed on a PC Intel Core i5-9400, 2.90 GHz with 8 GB of RAM. All models were implemented using ILOG CPLEX Optimization Studio 12.9.

A summary of the models that were tested is the following:

BNH: *Backward Node Hop-indexed* model defined by constraints (1), (2), (3), (15), (5), (6), (7), (8) and (9).
$sBNH$: *strong Backward Node Hop-indexed* model defined by constraints (1), (2), (3), (16) , (5), (6), (7), (8) and (9).
$gBNH$: *generalized Backward Node Hop-indexed* model defined by constraints (1), (2), (3), (17), (5), (6), (7), (8) and (9).
AH: *Arc Hop-indexed* model defined by constraints (18), (19), (20), (21), (22), (23), (24), (25), (26) and (27).
sAH: *strong Arc Hop-indexed* model defined by constraints (18), (19), (20), (28), (22), (23), (24), (25), (26) and (27).

And also for three of these models, $sBNH$, $gBNH$ and AH, we tested a version including the Generalized Subtour Elimination constraints of size two (10), $sBNH^*$, $gBNH^*$ and AH^*, respectively.

For this experiment we used a set of graphs already used in the computational experience in several previous works for the HSTP, *e.g.*, in the work by Gouveia [7]. In these graphs, the coordinates of $(n + 1)$ points were first randomly generated in a square grid. The cost of a candidate edge is then taken as the integer part of the Euclidean distance between the points defining the endpoints of the edge. The edge set E of the graph was then defined as follows: i) all the edges incident to the root node were included in E (this ensures that the problem has at least one feasible solution) and ii) the m least cost candidate edges not incident to the root, were also included in E. Thus, for each graph, we have $|E| = n + m$ for appropriate values of n and m, leading to fairly sparse graphs, which is typical in telecommunications networks. The original set contained two classes of graphs, depending on the location of the root node on the square grid: either located at the center (TC class) or at the corner (TE class). The two classes were used in our experiments.

The arc set is then build by considering every arc $(0, j), j \in \{1, \ldots, n\}$, and arcs $(i, j), (j, i)$, for every $\{i, j\}$ in the edge set. In order to reduce the size of each instance, we used a standard arc elimination test (as far as we know first used by Gouveia [6]), that consists in removing every arc $(i, j), i \neq 0$, such that $c_{ij} \geq c_{0,j}$. Table 2 shows the different values for n, m, $|E|$, $|A|$ and the number of arcs after the elimination test for the TC and TE classes (note that this reduction is small, due to the sparsity of the graphs and the way they were built).

n	60	80	100	120	160
m	150	200	250	300	400
$\|E\|$	210	280	350	420	560
$\|A\|$	360	480	600	720	960
$reduced\ \|A\|\ (TC)$	340	449	571	705	956
$reduced\ \|A\|\ (TE)$	355	476	595	720	959

TABLE 2 Instances graph sizes

For each graph, we tested four values for the number of required nodes: 25%, 50%, 75% and 100%, respectively of the total number of nodes (this last case corresponds to a hop-constrained spanning tree problem) and for the hop limit we tested, as in previous works, the following values $H = 3, 4, 5$, thus obtaining a total of 60 instances in each class.

4.1 | The LP performance of the models

Table 3 presents the gaps for the linear programming relaxations of the eight models. The format of the table is as follows: the first three columns define the instance size in terms of the number of nodes beside the root node (n), the number of required nodes beside the root node ($|R_1|$) and the hop limit (H). The following eight columns contain the LP gaps (in percentage) for the TC class of instances and the next eight columns contain the LP gaps for the TE class of instances. Figures 3 and 4 report the same results in the form of a performance profile graph for TC and TE instances, respectively. For each model, a curve represents the number of instances for which the gap is lower than a given value. The higher the curve, the better. A few observations can be derived from the reported results.

- The performance profile graphs clearly show that we can cluster the models in three groups: the arc models sAH, AH^* and AH, in this order, are the best ones, a second group is composed of $gBNH^*$, $gBNH$ and $sBNH^*$, and finally $sBNH$ and BNH are clearly the worst models with regards to LP bounds.
- As expected, TE instances have worse gaps than the TC instances although this observation is less evident in larger instances. This difference in the two classes is also seen later, when we report the CPU times to to obtain the optimal solutions (see Table 4).

			TC instances								TE instances							
n	$\|R_1\|$	H	BNH	$sBNH$	$sBNH^*$	$gBNH$	$gBNH^*$	AH	AH^*	sAH	BNH	$sBNH$	$sBNH^*$	$gBNH$	$gBNH^*$	AH	AH^*	sAH
		3	51	51	17	15	14	10	8	8	83	83	61	54	54	41	35	35
	7	4	52	52	18	26	18	18	14	13	82	82	57	65	57	50	47	45
		5	53	53	18	26	18	22	17	15	81	81	53	64	53	53	48	46
		3	55	55	27	24	23	19	16	16	70	70	51	49	49	27	25	23
	15	4	54	54	25	32	25	23	19	19	69	69	47	54	47	39	36	34
60		5	53	53	23	31	22	26	20	19	67	67	41	52	41	41	37	35
		3	41	41	26	23	23	12	11	11	57	57	44	44	43	20	18	15
	30	4	36	36	19	21	18	12	11	9	52	52	35	41	35	25	22	19
		5	33	33	14	19	14	13	10	9	49	44	29	37	29	27	23	20
		3	23	23	23	23	23	6	6	4	40	39	39	39	39	14	14	10
	60	4	19	19	16	17	16	8	7	5	38	38	32	34	31	19	18	14
		5	14	14	10	13	10	7	7	3	32	32	25	31	25	20	18	15
		3	63	63	29	24	24	19	16	16	82	82	66	62	62	43	38	37
	10	4	61	61	28	32	27	24	21	19	81	81	62	68	62	53	51	48
		5	60	60	26	31	26	27	23	22	80	80	60	68	60	59	55	53
		3	49	49	28	24	24	15	13	13	71	71	55	52	52	31	27	26
	20	4	46	46	26	30	25	21	18	18	67	67	48	53	48	39	36	34
80		5	44	44	23	29	23	23	20	19	67	67	47	53	47	45	42	40
		3	39	39	27	26	26	11	10	9	60	60	48	48	47	24	22	20
	40	4	35	35	22	25	22	16	15	13	55	55	39	44	39	29	27	24
		5	31	31	18	22	18	16	15	13	52	52	35	42	35	32	29	26
		3	23	23	23	23	23	7	7	6	45	45	45	45	45	17	17	14
	80	4	17	17	17	17	17	9	9	8	39	39	35	37	35	21	20	17
		5	12	12	12	12	12	9	8	6	34	34	29	34	29	23	22	18
		3	67	67	38	33	31	26	22	22	81	81	64	60	60	42	37	32
	12	4	64	64	34	38	33	30	27	26	80	80	60	66	60	51	47	44
		5	63	63	31	36	31	32	28	27	80	80	58	65	58	57	52	50
		3	50	50	33	30	29	19	16	15	71	71	55	54	54	32	29	26
	25	4	46	46	28	32	27	23	20	19	67	67	46	52	46	37	34	31
100		5	42	42	22	29	22	23	19	18	67	67	45	53	45	44	40	38
		3	37	37	26	24	24	11	10	9	60	60	46	46	46	23	21	18
	50	4	30	30	19	21	19	12	11	10	55	55	37	42	37	28	25	23
		5	25	25	14	17	13	12	10	8	52	52	32	39	32	30	26	24
		3	22	22	22	22	22	7	7	5	47	47	47	47	47	17	17	14
	100	4	15	15	15	15	15	9	9	7	43	43	38	40	38	22	22	18
		5	10	10	10	10	10	8	8	6	36	36	30	35	30	23	22	18
		3	76	76	55	51	51	37	32	31	86	86	73	70	70	51	45	43
	15	4	77	77	52	60	52	46	43	40	86	86	72	77	71	64	61	57
		5	77	77	51	61	51	52	47	44	85	85	67	74	67	66	62	59
		3	71	69	47	44	43	29	24	22	82	79	66	64	63	41	37	32
	30	4	69	69	42	51	42	34	31	29	80	80	61	68	61	50	46	43
120		5	68	68	39	51	39	41	34	32	78	78	58	66	58	56	52	49
		3	57	55	43	41	41	20	19	17	72	70	61	60	60	32	30	25
	60	4	55	55	35	43	35	26	24	21	68	68	54	59	54	41	38	34
		5	54	53	31	42	31	31	26	23	66	66	49	57	49	46	43	40
		3	36	35	35	35	35	13	13	10	49	48	48	48	48	20	20	14
	120	4	35	35	29	31	28	17	16	13	46	46	42	44	42	27	26	23
		5	31	31	23	29	23	19	17	13	43	43	37	42	37	31	30	27
		3	80	79	62	59	58	36	31	30	85	81	69	66	66	39	35	30
	20	4	77	77	55	61	53	43	39	34	84	84	69	73	68	56	52	45
		5	77	76	52	61	52	50	45	42	82	82	64	70	64	58	55	49
		3	77	76	61	60	59	36	32	30	83	80	70	68	68	40	37	34
	40	4	74	74	54	61	54	44	40	35	81	81	67	72	67	54	51	45
160		5	73	72	50	60	50	48	43	39	78	78	61	69	61	57	53	48
		3	65	62	51	51	50	24	21	19	72	71	64	63	63	31	29	24
	80	4	63	63	44	51	43	33	29	21	73	73	61	65	61	47	44	38
		5	61	61	42	52	41	38	33	25	69	69	54	61	54	48	46	41
		3	46	43	43	43	43	13	13	9	55	53	53	53	53	18	17	12
	160	4	50	50	41	45	41	26	24	17	59	59	54	56	54	36	35	28
		5	45	45	35	43	35	29	26	18	54	54	47	53	47	40	38	32

TABLE 3 LP gaps (%) for the TC and TE instances (LP CPU times are less than 2 sec.)

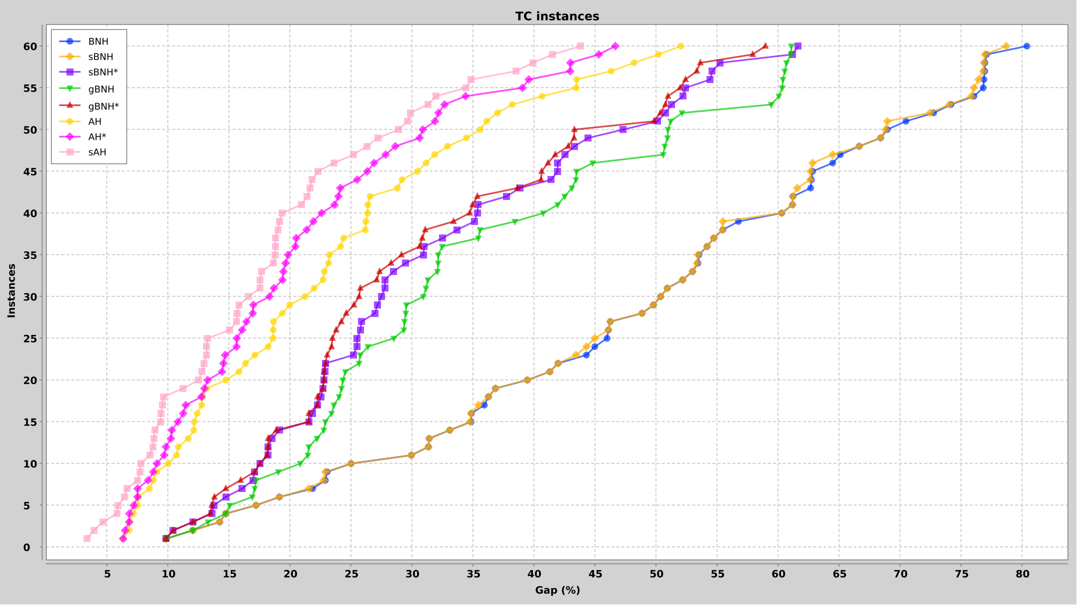

FIGURE 3 LP gaps (%) for the TC instances

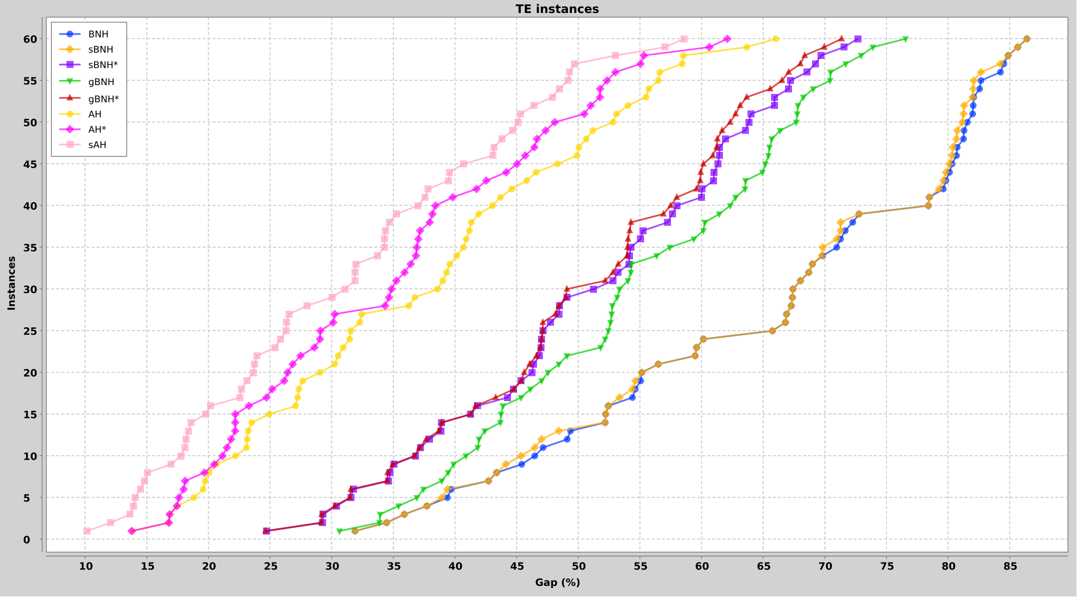

FIGURE 4 LP gaps (%) for the TE instances

- As expected, the node-based models produce LP bounds that are worse than the ones produced by the arc-based models. We observe that there is a clear improvement in LP bounds from the $sBNH$ model to the $gBNH$ model (on average 14% and 10% decrease on the TC and TE instances, respectively). This improvement is more significant in the case where not all of the nodes are required nodes. The inclusion of the generalized subtour elimination constraints of size two (10) in these models further reduces the gaps, but in this case the differences between the lower bounds of the enhanced models, $sBNH^*$ and $gBNH^*$, are smaller indicating that the effect of these inequalities is more effective on the weaker model.
- Comparing the node-based *versus* arc-based models, the weakest arc-based model, AH, outperforms the strongest node-based model, $gBNH$, in every instance (on average 11% and 17% decrease on the TC and TE instances, respectively). This observation still remains when we add the constraints (10) to these two models and it is more relevant for the larger instances in both classes, TC and TE.
- When comparing the two arc-based models, we observe that the stronger one outperforms the weaker one in every instance, with a decrease of 5% on average, most significantly for the larger instances. Adding constraints (10) to the weakest arc-based model reduces the average gap difference between the two models. Note also that the corresponding CPU times and the results of the next section indicate that even after this reduction, the strong arc model is preferable to the weak arc model with the subtour elimination constraints (10).

4.2 | Obtaining the optimal (integer) solutions

Although the LP values are important indicators for an overall comparison between all the models in our experiment, they are not sufficient to allow us to assess what is the best (faster) model to obtain the optimal integer solution. Since other factors need also to be considered, *e.g*, the size of the models as well as unknown factors of the ILP package used to solve the instances. In Table 4 we provide the CPU times (in seconds) to obtain the optimal integer solutions taken from all the models described before (the time limit was set to one hour). The format of the table is identical to the one in Table 3. The designation "O.M." indicates a model that could not be solved due to an "Out of Memory" issue and the designation "T.L." indicates a model that reached the "Time Limit" of one hour, before obtaining an optimal solution (or proving the optimality of the best found solution). Performance profile graphs for the same results are presented in Figures 5 and 6, showing the number of instances solved within a given CPU time (on a logarithmic time scale). Again, a higher curve corresponds to a better model.

As in previous works, the TC class instances are easier to solve rather than the TE class instances. Since the size of the models strongly depends on the hop limit, it is also expected that the CPU times increase when the value of H increases. Also, although for $H = 3$, the CPU times are insignificant, for $H = 5$ the CPU times are significantly larger, specially for the node-based models.

From the performance profile graphs in Figure 5 and 6, we can observe the following interesting fact: the addition of inequalities (10) to a model ($sBNH^*$, $gBNH^*$ or AH^*), although contributing to a reasonable improvement in the corresponding LP gaps, does not necessarily lead to models with better CPU times to obtain the optimal solutions. In fact, the performance profile graphs for CPU times show that the line corresponding to an enhanced model ($sBNH^*$, $gBNH^*$ and AH^*) is below the line corresponding to the original model without enhancements most of the time (this is more evident for models AH and AH^*). However, when comparing an enhanced model versus the corresponding original model, the increase in CPU times is usually not greater than 5 minutes and there are even some cases where the original model did not find the optimal solution within the time limit whereas the enhanced model did.

The comparison between the models AH, AH^* and sAH is also interesting since it allows us to provide some model improvement insights. As specified before, the model AH^* is obtained from the model AH by adding a set of

			TC instances								TE instances							
n	$\|R_1\|$	H	BNH	$sBNH$	$sBNH^*$	$gBNH$	$gBNH^*$	AH	AH^*	sAH	BNH	$sBNH$	$sBNH^*$	$gBNH$	$gBNH^*$	AH	AH^*	sAH
60	7	3	1	0	0	0	0	0	1	0	1	1	3	1	1	0	1	0
		4	1	1	1	1	1	0	0	0	10	12	15	13	28	1	6	1
		5	1	1	1	2	1	1	1	0	85	45	41	110	79	19	26	28
	15	3	1	1	1	1	0	1	1	0	2	2	3	2	2	1	1	1
		4	3	3	2	3	2	1	1	1	35	21	30	36	27	12	13	2
		5	7	6	3	18	8	3	4	0	55	73	64	195	112	28	102	25
	30	3	1	1	1	1	1	1	1	1	2	2	2	2	2	1	1	1
		4	2	3	3	3	4	1	1	1	6	4	14	15	16	2	3	1
		5	10	7	4	13	20	1	1	0	57	57	30	59	135	14	13	5
	60	3	1	1	1	1	1	1	1	0	2	1	1	2	1	1	1	1
		4	5	2	4	4	8	1	1	1	3	2	2	5	5	2	2	1
		5	6	4	5	18	18	2	1	0	40	19	14	93	138	16	23	5
80	10	3	1	1	2	1	1	1	1	1	1	1	2	1	1	1	1	1
		4	4	5	2	3	4	1	1	1	8	8	30	85	25	2	3	1
		5	10	11	9	18	9	3	4	0	53	48	31	135	55	5	69	21
	20	3	1	1	2	1	1	1	1	1	1	1	5	3	3	1	1	1
		4	3	2	6	4	7	1	1	1	12	17	105	70	45	4	4	2
		5	13	12	25	33	16	6	4	0	292	157	160	235	132	68	240	37
	40	3	2	1	2	3	3	1	1	1	3	2	3	2	3	1	1	1
		4	10	25	32	63	69	2	2	1	12	16	29	41	27	3	3	2
		5	32	93	63	98	331	11	14	0	126	216	250	359	543	16	63	75
	80	3	3	2	2	2	1	1	1	1	2	2	1	1	2	1	1	1
		4	6	7	7	16	41	3	2	2	13	12	4	22	31	3	3	3
		5	43	10	29	166	447	10	11	0	29	42	59	758	1229	72	16	8
100	12	3	2	4	4	9	8	1	1	1	2	3	7	5	8	1	1	1
		4	29	76	79	75	57	15	14	2	119	523	499	715	287	5	19	10
		5	541	434	325	433	401	195	171	24	*T.L.*	*T.L.*	2059	3255	2048	*T.L.*	3479	418
	25	3	2	6	6	6	6	1	1	1	7	6	23	8	10	2	2	1
		4	23	88	86	84	82	6	5	2	102	413	413	230	1059	14	7	3
		5	203	126	101	86	78	29	27	4	*O.M.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	3571	*T.L.*	1438
	50	3	5	5	5	6	6	1	1	1	5	6	11	11	9	3	2	2
		4	21	76	76	53	50	5	4	2	174	1358	1385	956	608	18	6	4
		5	73	187	138	166	156	15	14	3	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	1132	*T.L.*	138
	100	3	4	3	3	5	5	2	2	2	7	4	5	9	8	2	2	2
		4	31	30	31	16	17	7	7	4	82	22	22	107	30	38	17	25
		5	2750	1098	974	*T.L.*	*T.L.*	70	60	36	544	385	387	1824	400	933	1713	178
120	15	3	3	3	4	10	12	1	2	1	3	2	5	19	20	2	2	2
		4	40	64	86	141	266	4	9	3	1379	1665	2054	*T.L.*	*T.L.*	9	16	6
		5	2274	2160	1188	2153	*T.L.*	190	709	145	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	506	*T.L.*	128
	30	3	5	8	10	15	27	2	2	1	13	11	26	67	46	2	5	2
		4	168	460	433	*T.L.*	660	8	31	5	118	1175	2811	*T.L.*	2621	15	205	18
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	316	*T.L.*	128	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	918	*T.L.*	182
	60	3	6	9	9	25	17	2	2	2	8	4	11	46	27	3	2	2
		4	464	124	310	1853	1609	6	16	5	321	448	265	2787	3533	18	41	6
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	719	*T.L.*	219	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	2076	474
	120	3	6	5	4	15	32	2	3	2	12	6	7	42	32	3	3	3
		4	183	94	44	392	188	10	21	6	38	43	26	98	138	18	40	27
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	511	1714	98	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*
160	20	3	8	4	8	25	21	3	4	2	12	8	23	48	55	3	4	2
		4	1035	361	442	700	398	9	50	6	42	54	1020	622	641	31	61	6
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	3017	*T.L.*	190	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	200	293	56
	40	3	107	154	346	443	446	4	8	3	20	24	31	99	76	4	4	3
		4	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	327	1154	94	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	359	1126	171
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	2787	*T.L.*	493	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	285
	80	3	212	111	181	305	458	14	21	6	151	94	111	224	394	8	10	7
		4	746	*T.L.*	*T.L.*	*T.L.*	*T.L.*	300	518	103	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	66	147	182
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*
	160	3	47	50	45	111	129	7	7	13	39	37	53	84	109	15	17	10
		4	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	329	*T.L.*	564	651	455	750	*T.L.*	2379	72	258	77
		5	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	1554	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*	*T.L.*

TABLE 4 CPU times (sec.) for solving the IP models

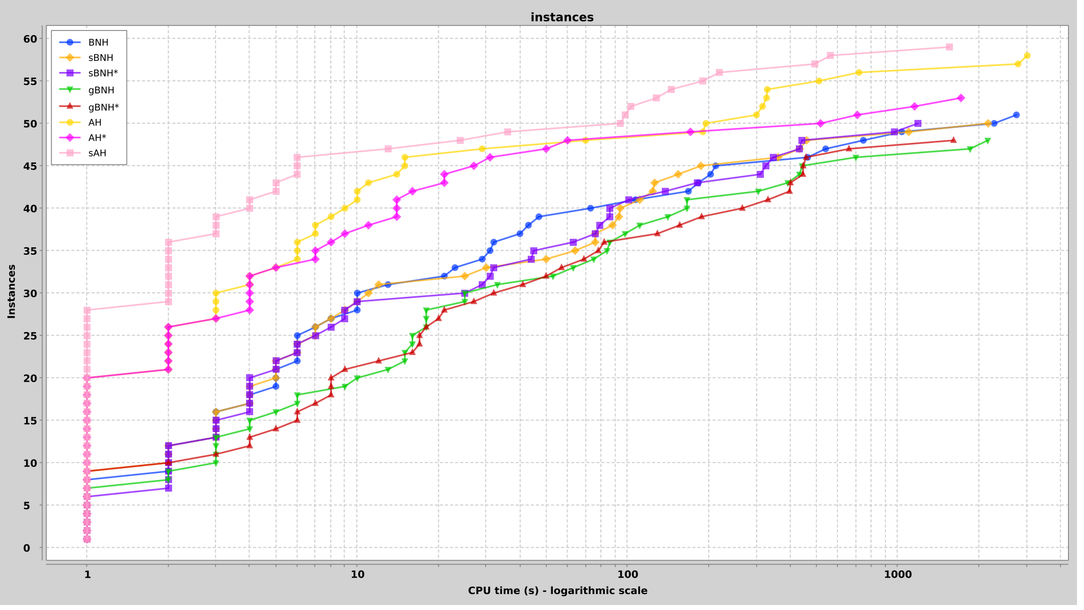

FIGURE 5 CPU times (sec.) for the TC instances

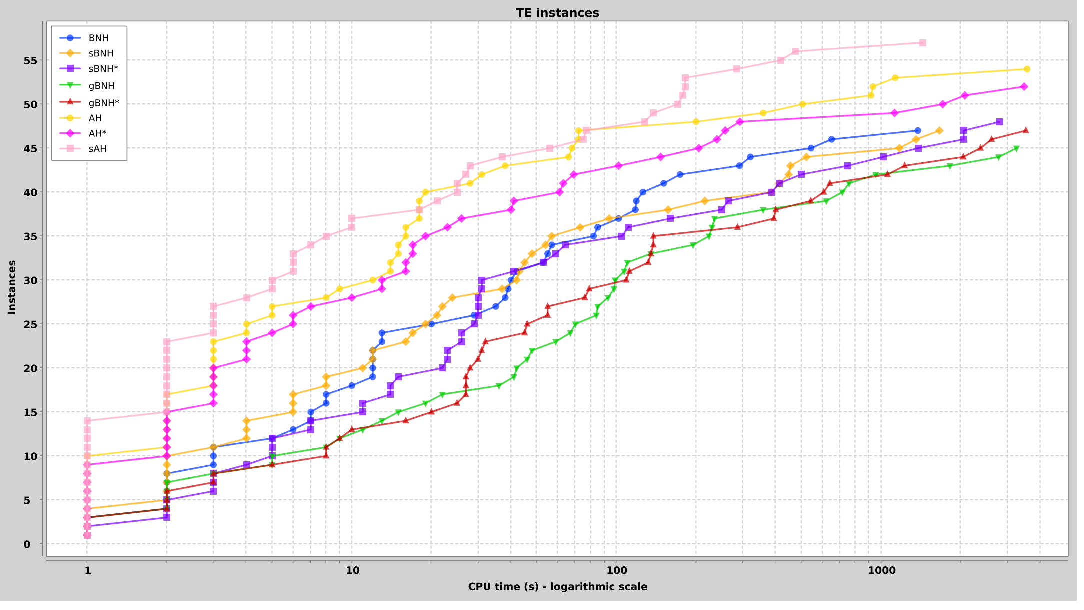

FIGURE 6 CPU times (sec.) for TE instances

constraints ((10) in this case). The model sAH is obtained from strengthening (lifting) a set of inequalities in model AH. As proved in Proposition 6, the model sAH implies the extra constraints in model AH^*. This explains the improvements in the lower bounds when going from model AH to model AH^* and then to model sAH. However, these relations are not necessarily "propagated" to the integer CPU times since model AH^* has more constraints than model AH, while model sAH is of the same size as model AH. The main conclusion from this analysis is that whenever possible, strengthening a constraint (which implies the original constraint plus a new set), besides strengthening the LP bounds, leads to a more efficient model to be solved, at least when compared with the original model plus the extra set of constraints.

As a conclusion, we observe that the best three models are the arc-based models with sAH clearly dominating, and AH better than AH^* despite the better bounds of AH^*. It is also noteworthy to observe that $gBNH^*$, despite being the best node model in terms of LP gap, becomes the worst one for TC instances, and the next to worst one for TE instances, with respect to CPU times for solving the problem to optimality.

5 | CONCLUSIONS

In this paper we have contextualized the linear programming relaxation of a *hop-indexed node-based model* (and also of a variant including a large set of generalized inequalities) in terms of a simple *hop-indexed arc-based model*. More precisely we have shown that the linear programming relaxation of the first model is implied by the linear programming relaxation of the second model. We observe that the result "arc model implies node model" is not surprising due to the equalities relating the arc variables with the node variables. This theoretical dominance was then evaluated, in practice, with the results taken from a computational experiment. The results indicate that despite using more variables, the arc-based models might be preferable to the node-based models when solving instances of the HTSP. This might be explained by the difference in gap values reported in the computational experiment. There are two points worth discussing. First, the time-dependent models can be viewed as models in a layered graph where different layers correspond to different positions. Also the inequalities (28) defining the strong arc model can be viewed as "simple" cut-set inequalities in the layered graph (see, e.g., the work by Gouveia, Leitner and Ruthmair [10]) and one wonders what inequalities in the space of the node variables are implied by the more general cut-set inequalities. Second, it would be interesting to try to "enlarge" the relations established in this paper by adding relations or non-dominance relations with other node-based models such as, for instance, the "weak" LP based Miller-Tucker-Zemlin model. Despite having a weak LP bound, this model is very compact and can allow the determination of optimal integer solutions with reasonable computing times, with current ILP packages, in cases where theoretically stronger models might fail, due to a large number of variables (such as in the time-dependent model).

references

[1] Botton, Q., Fortz, B. and Gouveia, L. (2015) On the hop-constrained survivable network design problem with reliable edges. *Computers & Operations Research*, **64**, 159–167.

[2] Botton, Q., Fortz, B., Gouveia, L. and Poss, M. (2013) Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, **25**, 13–26.

[3] Costa, A. M., Cordeau, J.-F. and Laporte, G. (2009) Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. *Networks*, **53**, 141–159.

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

[5] De Boeck, J. and Fortz, B. (2018) Extended formulation for hop constrained distribution network configuration problems. *European Journal of Operational Research*, **265**, 488–502.

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

[7] — (1998) Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. *INFORMS Journal on Computing*, **10**, 180–188.

[8] — (1999) Using hop-indexed models for constrained spanning and Steiner tree models. In *Telecommunications Network Planning (editors: Sansò, B. and Soriano, P.)*, 21–32. Springer.

[9] Gouveia, L., Leitner, M. and Ljubić, I. (2014) Hop constrained Steiner trees with multiple root nodes. *European Journal of Operational Research*, **236**, 100–112.

[10] Gouveia, L., Leitner, M. and Ruthmair, M. (2019) Layered graph approaches for combinatorial optimization problems. *Computers & Operations Research*, **102**, 22–38.

[11] Gouveia, L., Simonetti, L. and Uchoa, E. (2011) Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming*, **128**, 123–148.

[12] Leitner, M. (2016) Layered graph models and exact algorithms for the generalized hop-constrained minimum spanning tree problem. *Computers & Operations Research*, **65**, 1–18.

[13] Ljubić, I. and Gollowitzer, S. (2013) Layered graph approaches to the hop constrained connected facility location problem. *INFORMS Journal on Computing*, **25**, 256–270.

[14] Mahjoub, A. R., Simonetti, L. and Uchoa, E. (2013) Hop-level flow formulation for the survivable network design with hop constraints problem. *Networks*, **61**, 171–179.

[15] Sinnl, M. and Ljubić, I. (2016) A node-based layered graph approach for the Steiner tree problem with revenues, budget and hop-constraints. *Mathematical Programming Computation*, **8**, 461–490.

[16] Voß, S. (1999) The Steiner tree problem with hop constraints. *Annals of Operations Research*, **86**, 321–345.