On the Hop Constrained Steiner Tree Problem with Multiple Root Nodes

Luis Gouveia[1,⋆], Markus Leitner[2], and Ivana Ljubić[3,⋆⋆]

[1] Departamento de Estatística e Investigação Operacional - Centro de Investigação Operacional, Faculdade de Ciênçias, Universidade de Lisboa, Portugal
legouveia@fc.ul.pt

[2] Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria
leitner@ads.tuwien.ac.at

[3] Department of Statistics and Operations Research, University of Vienna, Austria
ivana.ljubic@univie.ac.at

Abstract. We consider a new network design problem that generalizes the Hop and Diameter Constrained Minimum Spanning and Steiner Tree Problem as follows: given an edge-weighted undirected graph whose nodes are partitioned into a set of root nodes, a set of terminals and a set of potential Steiner nodes, find a minimum-weight subtree that spans all the roots and terminals so that the number of hops between each *relevant node* and an arbitrary root does not exceed a given hop limit H. The set of relevant nodes may be equal to the set of terminals, or to the union of terminals and root nodes. This paper presents theoretical and computational comparisons of flow-based vs. path-based mixed integer programming models for this problem. Disaggregation by roots is used to improve the quality of lower bounds of both models. To solve the problem to optimality, we implement branch-and-price algorithms for all proposed formulations. Our computational results show that the branch-and-price approaches based on path formulations outperform the flow formulations if the hop limit is not too loose.

1 Introduction

We consider the Hop Constrained Minimum Steiner Tree Problem on a graph with Multiple Root nodes (HCSTPMR). Formally, we are given an undirected graph $G = (V, E)$, with node set V, edge set E, edge costs $c_e \geq 0$, $\forall e \in E$, and a hop limit $H \in \mathbb{N}$. The node set V is partitioned into the set of root nodes R, $|R| \geq 1$, a (potentially empty) set of terminal nodes $T \subset V \setminus R$, and the set of remaining nodes $S = V \setminus \{R \cup T\}$ that will be called *potential Steiner nodes.* Furthermore, we are given a set $T' \in \{T, R \cup T\}$, $T' \neq \emptyset$, of *relevant nodes* for which hop limits to all root nodes need to be considered.

⋆ The work of this author is supported by National Funding from FCT - Fundação para a Ciência e Tecnologia, under the project: PEst-OE/MAT/UI0152.

⋆⋆ The work of this author is supported by the APART fellowship of the Austrian Academy of Sciences.

A.R. Mahjoub et al. (Eds.): ISCO 2012, LNCS 7422, pp. 201–212, 2012.

A solution to the HCSTPMR is a Steiner tree $G' = (V', E')$ spanning all root and terminal nodes, i.e., $R \cup T \subseteq V'$, such that the hop-constraints are met for all pairs (u, v) with $u \in R$ and $v \in T'$. More precisely, if $P_{G'}(u, v) \subseteq E'$ denotes the edge set of the unique path from $u \in R$ to $v \in T'$ in G' and $l_{G'}(u, v) = |P_{G'}(u, v)|$ its length, then each feasible solution has to satisfy:

$$l_{G'}(u, v) \leq H, \quad \forall u \in R,\ \forall v \in T'.$$

The objective is to find a feasible solution $G^* = (V^*, E^*)$ yielding minimum total edge costs, i.e. $\min \sum_{e \in E^*} c_e$.

The HCSTPMR is NP-hard since it becomes the Hop Constrained Steiner Tree Problem (HCSTP) for $|R| = 1$ or $|T'| = 1$. Furthermore, we have the Diameter Constrained Steiner Tree Problem (DCSTP) with diameter equal to H if $T' = R$. The Hop and Diameter Constrained Minimum Spanning and Steiner Tree Problems have been studied by many authors, see e.g. [6] and [10] for recent contributions. The HCSTPMR which generalizes these well known network design problems has not been studied in the literature before.

Overview of the paper. In Section 2 we first consider undirected flow- and path-based mixed integer programming (MIP) formulations. Due to the hop-constraints imposed for terminals with respect to each of the root nodes $s \in R$, it is not obvious how to direct a feasible solution whenever $|R| > 1$. To overcome this disadvantage, in Section 3 we propose a model that considers $|R|$ directed models, one for each root $s \in R$, combined together with adequate coupling of the directed variables from each model with the undirected edge variables. We also compare the proposed formulations with respect to their quality of LP lower bounds. Section 4 provides implementation details of branch-and-price approaches that have been implemented for the proposed MIP formulations. Computational comparison is conducted in Section 5 where lower bounds of our MIP formulations are calculated and the overall performance of the branch-and-price approaches is compared on a set of publicly available benchmark instances.

Notation. By $\mathcal{P}_M$ we denote the convex hull of all feasible LP solutions of a MIP formulation M and by $\text{proj}_{\boldsymbol{a}^1,\ldots,\boldsymbol{a}^n}(\mathcal{P}_M)$, the orthogonal projection of the convex hull of LP solutions of M onto the space defined by variables $\boldsymbol{a^1}, \ldots, \boldsymbol{a^n}$.

2 Undirected Formulations

For the formulations considered throughout this section, for each $s \in R$ and $t \in T'$, $t \neq s$, we consider a pair (s, t) as a *commodity.* Our goal is to find an optimal solution that includes a path between s and t with at most H hops, for each commodity pair (s, t). To model a feasible solution $G' = (V', E')$ on G, we will use binary edge variables, x_e, that are set to one if $e \in E'$, and to zero, otherwise, for all $e \in E$. In addition, we will use binary node variables associated to potential Steiner nodes: y_v is set to one if $v \in V' \cap S$, and to zero, otherwise, for all nodes $v \in S$. Finally, $A = \{(i, j), (j, i) \mid \{i, j\} \in E\}$ denotes the set of bi-directed arcs in G.

2.1 An Undirected Multi-commodity Flow Formulation

Multi-commodity-flow-based formulations are one of the oldest approaches for modeling hop-constrained network design problems (see, e.g., [2]). Our formulation (1)–(10) to which we refer as UFlow^{B} uses continuous flow variables $f_{ij}^{st} \geq 0$, denoting the amount of flow of commodity (s,t) sent through an arc $(i,j) \in A$ for each $s \in R$, $t \in T' \setminus \{s\}$.

$$\min \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t.} \sum_{(i,j) \in A} f_{ij}^{st} - \sum_{(j,i) \in A} f_{ji}^{st} = \begin{cases} 1 & \text{if } i = s \\ 0 & \text{if } i \neq s,t \\ -1 & \text{if } i = t \end{cases} \quad \forall s \in R,\ \forall t \in T' \setminus \{s\},\ \forall i \in V \tag{2}$$

$$\sum_{(i,j) \in A} f_{ij}^{st} \leq H \qquad \forall s \in R,\ \forall t \in T' \setminus \{s\} \tag{3}$$

$$0 \leq f_{ij}^{st} + f_{ji}^{st} \leq x_e \qquad \forall s \in R,\ \forall t \in T' \setminus \{s\},\ \forall e = \{i,j\} \in E \tag{4}$$

$$x_e \leq y_i \qquad \forall e = \{i,j\} \in E, i \in S \tag{5}$$

$$\sum_{e \in E} x_e = |R| + |T| + \sum_{v \in S} y_v - 1 \tag{6}$$

$$\sum_{e = \{i,j\} \in E} x_e \geq 2 y_i \qquad \forall i \in S \tag{7}$$

$$0 \leq f_{ij}^{sp} + f_{ji}^{sq} \leq x_e \qquad \forall s \in R,\ \forall p,q \in T' \setminus \{s\},\ p \neq q,\ \forall e = \{i,j\} \in E \tag{8}$$

$$y_i \in \{0,1\} \qquad \forall i \in S \tag{9}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{10}$$

Constraints (2) are the classical flow conservation constraints while inequalities (3) are the hop-constraints restricting the maximum length of the path between each root $s \in R$ and each relevant terminal $t \in T'$. Constraints (4) are the undirected linking constraints between flow and edge variables. By modeling the problem using only constraints (2)-(4) and (10), one will obtain a subgraph of G that connects all roots and terminals, and respects all the hop limits, but which is not necessarily a tree. To ensure the tree structure of the feasible solution, additional linking constrains are added: inequalities (5) make sure that a node variable is set to one whenever an incident edge is taken into solution, and constraint (6) makes sure that the number of edges in the solution is one less than the number of nodes. Still, even the model (2)-(6) plus (9)-(10) does not ensure that the obtained solution is a tree, as illustrated by an example in Figure 1. For $H = 3$ and $T' = R \cup T$ no feasible solution exists. Without constraints (7), however, the model (2)-(6) plus (9)-(10) admits the solution given in Figure 1b. Therefore, inequalities (7) are necessary to ensure that the final solution contains only a single connected component. Finally, we also add *bidirectional commodity-pair forcing constraints* (8), see e.g. [3], to strengthen our model. Validity of these constraints follows from the fact that, for a given root node $s \in R$, there always exists an optimal solution in which flow of

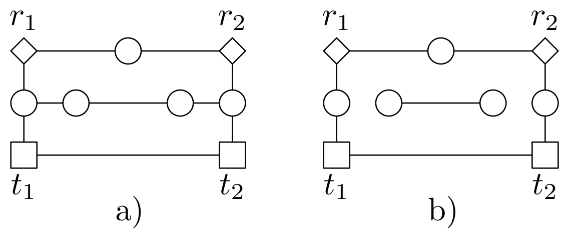

Fig. 1. a) An exemplary instance with $R = \{r_1, r_2\}$, $T = \{t_1, t_2\}$ infeasible for $H = 3$ and $T' = R \cup T$ and b) a solution feasible for UFlow without constraints (7)

two different commodities (s, p) and (s, q) will always be sent in the same direction along an edge.

In our theoretical comparison and computational experiments we also consider a variant without constraints (8) to which we refer as UFlow. Following from what is known for problems without hop-constraints, cf. [3], one can easily find instances showing that UFlow^{B} is strictly stronger than UFlow.

2.2 An Undirected Path Formulation

We now propose a path-based formulation by considering the set of all directed hop constrained paths $\mathcal{W}_{st} \subseteq 2^A$, i.e. $l_G(p) \leq H$, $\forall p \in \mathcal{W}_{st}$, from each root $s \in R$ to each relevant terminal $t \in T' \setminus \{s\}$. For each commodity pair (s, t), $s \in R$, $\forall t \in T' \setminus \{s\}$, we introduce an exponential number of path variables $0 \leq \lambda_p^{st} \leq 1$, $\forall p \in \mathcal{W}_{st}$. Constraints (2)–(4) of model UFlow^{B} are now replaced by inequalities (11)–(13). Further replacing inequalities (8) by the bidirectional commodity-pair forcing constraints (14) yields model UPath^{B}.

$$\sum_{p \in \mathcal{W}_{st}} \lambda_p^{st} = 1 \qquad \forall s \in R,\ \forall t \in T' \setminus \{s\} \quad (11)$$

$$\sum_{p \in \mathcal{W}_{st} : (i,j) \in p \vee (j,i) \in p} \lambda_p^{st} \leq x_e \qquad \forall s \in R,\ \forall t \in T' \setminus \{s\}, \forall e = \{i, j\} \in E \quad (12)$$

$$\lambda_p^{st} \geq 0 \qquad \forall s \in R,\ \forall t \in T' \setminus \{s\},\ \forall p \in \mathcal{W}_{st} \quad (13)$$

$$\sum_{\substack{p \in \mathcal{W}_{st}: \\ (i,j) \in p}} \lambda_p^{su} + \sum_{\substack{p \in \mathcal{W}_{sv}: \\ (j,i) \in p}} \lambda_p^{sv} \leq x_e \qquad \forall s \in R,\ \forall u, v \in T' \setminus \{s\},\ u \neq v,\ \forall e = \{i, j\} \in E \quad (14)$$

Constraints (11) ensure that each terminal is connected to each root node by a feasible path while constraints (12) are the linking constraints to corresponding edge variables. Notice that replacing equations (11) by "$\geq$" inequalities ensuring that each terminal is connected to each root by at least one feasible path would also yield a valid model. Since the latter is often computationally advantageous we use this variant in our computational experiments while we stick to the equality constraints in the following theoretical comparison of our models.

Since the number of path variables may be exponentially large, we apply column generation to solve the LP relaxation of this model (cf. Section 4).

As for $\mathrm{UFlow}^{\mathrm{B}}$ we can obtain a weaker model UPath with significantly less constraints by removing inequalities (14). By similar arguments as for the single root case, one can easily obtain that $\mathrm{UPath}^{\mathrm{B}}$ is strictly stronger than UPath.

One can show the following result:

Lemma 1. *Path-based models (*UPath, $\mathrm{UPath}^{\mathrm{B}}$*) are strictly stronger than their flow-based counterparts (*UFlow, $\mathrm{UFlow}^{\mathrm{B}}$*), respectively. However, models* UPath *and* $\mathrm{UFlow}^{\mathrm{B}}$ *are incomparable.*

The proofs of the results stated in Lemma 1 are similar to ones described in [6] for the single root case, where the authors show that a path model is equivalent to a compact model where the hop-constrained subproblem is modeled as an unconstrained path problem in a layered graph. The fact that the latter model is then strictly stronger than the flow model follows immediately. These proofs as well as layered graph subproblem model are easily adapted for the case with multiple roots. For simplicity we omit this from here.

3 Disaggregating Design Variables by Root Nodes

One of the difficulties when modeling a hop constrained problem with multiple root nodes is that it is far from obvious how to "direct the model", since a feasible solution basically consists of a "core" subtree spanning all the roots (and probably some of terminals) and the remaining subtrees attached to it. The core subtree cannot be directed, since each root defines its own set of hop-constraints. In order to use the strength of "directing the model" we will model the problem in a different way. Each feasible solution can be seen as the union of $|R|$ hop constrained subtrees, each one with a hop-constraint associated to the path between the corresponding root and each node in T'. Thus, we consider a new model with variables associated to each one of these trees as well as the original design variables x_e to guarantee that each rooted tree solution maps into the same tree. The advantage of this approach, is that we can direct each one of the rooted models and obtain a model with a stronger LP relaxation. We present next the model for the whole problem containing the directed version of the rooted tree models.

For each $s \in R$ we consider directed arc variables $a^s_{ij} \in \{0,1\}$, $\forall (i,j) \in A$. Variable a^s_{ij} is equal to one if arc (i,j) is part of the hop-constrained arborescence rooted at $s \in R$, and to zero, otherwise. Each solution must then contain $|R|$ directed Steiner arborescences and installation costs of edges $e = \{i,j\}$ need to be paid whenever either arc (i,j) or arc (j,i) is used by at least one of them. Hence, the following coupling constraints state the connection between a^s_{ij} and x_e variables:

$$a^s_{ij} + a^s_{ji} = x_e \quad \forall s \in R,\ \forall e = \{i,j\} \in E \tag{15}$$

Furthermore, in model UFlow we need to replace (4) by

$$0 \le f^{st}_{ij} \le a^s_{ij} \quad \forall s \in R,\ \forall t \in T' \setminus \{s\},\ \forall (i,j) \in A \tag{16}$$

to obtain model UFlow$^{\mathrm{D}}$ while we substitute (12) by

$$\sum_{p \in \mathcal{W}_{st}:(i,j)\in p} \lambda_p^{st} \le a_{ij}^s, \;\; \forall s \in R, \;\; \forall t \in T' \setminus \{s\}, \forall (i,j) \in A \qquad (17)$$

in model UPath yielding model UPath$^{\mathrm{D}}$.

Notice that the flow or path variables used to model different arborescences may use different edge sets, if $T' \neq T \cup R$. We obtain tighter LP bounds, by nevertheless using equations instead of inequalities in constraints (15), cf. [11].

Lemma 2. *Disaggregated models are equally strong as the corresponding "bidirectional" models, i.e.* $\mathrm{proj}_x(\mathcal{P}_{\mathrm{UFlow^D}}) = \mathrm{proj}_x(\mathcal{P}_{\mathrm{UFlow^B}})$ *and* $\mathrm{proj}_x(\mathcal{P}_{\mathrm{UPath^D}}) = \mathrm{proj}_x(\mathcal{P}_{\mathrm{UPath^B}})$.

Proof. We will show only the first equality, the second one can be proved analogously. First assume that $x \in \mathcal{P}_{\mathrm{UFlow^D}}$. Then for each edge $e = \{i,j\} \in E$ and each root $s \in R$ we have $x_e = a_{ij}^s + a_{ji}^s$. Furthermore, for each pair of terminals $p, q \in T' \setminus \{s\}$, $p \neq q$, $a_{ij}^s \ge f_{ij}^{sp}$ and $a_{ji}^s \ge f_{ji}^{sq}$. Hence $x_e = a_{ij}^s + a_{ji}^s \ge f_{ij}^{sp} + f_{ji}^{sq}$ and thus $x \in \mathcal{P}_{\mathrm{UFlow^B}}$.

Now, consider $x \in \mathcal{P}_{\mathrm{UFlow^B}}$. For an edge $e = \{i,j\} \in E$ and root $s \in R$, let terminals $p(s), q(s) \in T' \setminus \{s\}$ be defined such that $p(s) = \mathrm{argmax}_{t \in T' \setminus \{s\}}\{f_{ij}^{st}\}$ and $q(s) = \mathrm{argmax}_{t \in T' \setminus \{s\}}\{f_{ji}^{st}\}$. Then $f_{ij}^{sp(s)} + f_{ji}^{sq(s)} \le x_e$ holds either due to constraints (4) (if $p = q$), or due to constraints (8) (otherwise). Consider now root $s' = \mathrm{argmax}_{s \in R} f_{ij}^{sp(s)} + f_{ji}^{sq(s)}$. We define $a_{ij}^{s'} = f_{ij}^{s'p(s')}$ and $a_{ji}^{s'} = f_{ji}^{s'q(s')}$. For the remaining roots $s \neq s'$, we set $a_{ij}^s = f_{ij}^{sp(s)}$ and $a_{ji}^s = a_{ij}^{s'} + a_{ji}^{s'} - f_{ij}^{sp(s)}$. Hence, for all $e \in E$ and $s \in R$ constraints $x_e = a_{ij}^s + a_{ji}^s$ are satisfied. Capacity constraints (16) hold as well and we have $x \in \mathcal{P}_{\mathrm{UFlow^D}}$. □

For both models we can additionally add the following *disaggregated node-degree constraints*:

$$\sum_{(j,i)\in A} a_{ji}^s = \begin{cases} y_i, & i \in S \\ 0, & i = s \\ 1, & \text{else} \end{cases} \;\forall s \in R, \; \forall i \in V \quad \text{and} \quad \sum_{(i,j)\in A} a_{ij}^s \ge y_i \quad \forall s \in R, \; \forall i \in S \qquad (18)$$

We will refer to the resulting disaggregated formulations with constraints (18) by UFlow$^{\mathrm{DI}}$ and UPath$^{\mathrm{DI}}$.

Lemma 3. *Formulations* UFlow$^{\mathrm{DI}}$ *and* UPath$^{\mathrm{DI}}$ *are strictly stronger than formulations* UFlow$^{\mathrm{D}}$ *and* UPath$^{\mathrm{D}}$*, respectively. Furthermore, constraints* (6) *and* (7) *are redundant when constraints* (18) *are included.*

Proof. To show that UPath$^{\mathrm{DI}}$ is strictly stronger than UPath$^{\mathrm{D}}$, we consider Figure 2. Figure 2a shows a feasible LP-solution of UPath$^{\mathrm{D}}$ for $H = 3$ and $T' = T$. Figures 2b, 2c, and 2d show the values of disaggregated arc variable values $a^{r_1}, a^{r_2}, a^{r_3}$, respectively. Figure 2e resembles the only possibility to orient this solutions with r_2 as root node such that the disaggregated node-degree

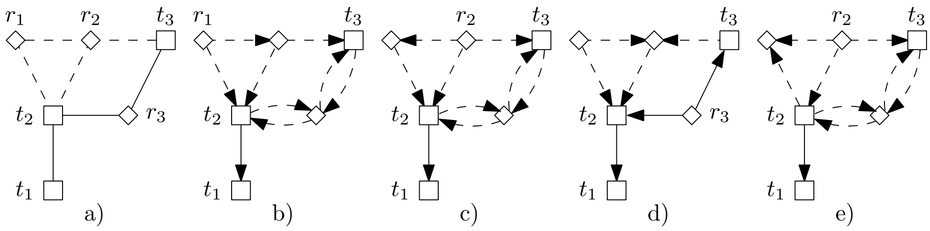

Fig. 2. a) A feasible solution to $\mathcal{P}_{\text{UPath}^{\text{D}}}$. Solid and dashed edges correspond to $x_e = 1$ and $x_e = 1/2$, respectively. Values of b) a^{r_1}, c) a^{r_2}, d) a^{r_3} for the same feasible solution. e) Values of a^{r_2} that satisfy additional disaggregated node-degree constraints, but that violate path constraints (11) for $s = r_2$, $t = t_1$, and $H = 3$. Solid and dashed arcs indicate values of 1 and 0.5, respectively.

constraints are satisfied for all $v \in V$. In Figure 2e, however, there is only one feasible path from r_2 to t_1 with maximum value of 0.5 since $a^{r_2}_{r_2 t_2} = 0.5$. Hence, this solution is not contained in $\mathcal{P}_{\text{UPath}^{\text{DI}}}$. Similar examples can be constructed to show that UFlow^{DI} is strictly stronger than UFlow^{D}.

To show that constraints (6) are implied, consider an arbitrary $s \in R$:

$$\sum_{e \in E} x_e = \sum_{(i,j) \in A} a^s_{ij} = \sum_{(i,j) \in A: j \in (R \cup T)} a^s_{ij} + \sum_{(i,j) \in A: j \in S} a^s_{ij} = |R| + |T| - 1 + \sum_{j \in S} y_j$$

For potential Steiner nodes $i \in V \setminus (R \cup T)$ using constraints (18) we obtain inequalities (7) as follows:

$$\sum_{e = \{i,j\} \in E} x_e = \sum_{(j,i) \in A} a^s_{ji} + \sum_{(i,j) \in A} a^s_{ij} \geq y_i + y_i = 2y_i$$

4 Branch-and-Price Algorithms

The MIP formulations considered throughout this paper exhibit a very large (flow-based models) or even exponential (path-based models) number of variables, and henceforth, decomposition-based approaches are inevitable when it comes to solving these models in practical applications. Column generation, or more general branch-and-price algorithms, are a common way to approach path-based models. On the other hand, flow-based formulations are frequently approached by Lagrangian relaxation [8] or Benders decomposition [5]. In this paper we propose to solve both types of formulations using branch-and-price, i.e. by embedding column generation into branch-and-bound.

Column generation for the flow-based formulations. Applications of column generation to flow-based models have been described recently in [7,12]. Pricing in these models is done on the set of design variables x_e, $e \in E$. Notice that a variable x_e set to zero, implies that all flow variables corresponding to the same edge will be zero as well, which explains why, in some cases, we may benefit from solving a smaller LP, in which only a subset of design variables is considered.

Since finding a feasible HCSTPMR solution is not an easy task, we initialize the restricted master problem as follows: we enlarge the input graph G by a set of dummy edges (of very large cost) between the nodes defining commodity pairs and all root nodes, whenever such an edge does not exist in G. The LP is then initialized using the edges of the subgraph induced by the set of nodes $R \cup T$. In the pricing subproblem, we explicitly calculate reduced costs of non-active dual variables. Notice that this can be done in polynomial time, since we are solving the compact flow-based models. All variables with negative reduced costs are inserted into the restricted master problem at once, and the process is repeated as long as edges with negative reduced costs can be found.

Column generation for the path-based formulations. For each $s \in R$, $t \in T' \setminus \{s\}$ and each $e = \{i,j\} \in E$, let μ_{st} and π_{ij}^{st} be the dual variables associated to constraints (11) and (12), respectively. Then, for UPath the reduced costs $\bar{c}_p$ for variable λ_p^{st} corresponding to a path $p \in \mathcal{W}_{st}$, $s \in R$, $t \in T' \setminus \{s\}$ are defined as

$$\bar{c}_p = -\mu_{st} - \sum_{e=\{i,j\} \in E} \pi_{ij}^{st}. \tag{19}$$

It is not difficult to see that the variable yielding minimum reduced costs can be obtained by solving a hop constrained shortest path problem (HCSPP) between each root $s \in R$ and each relevant terminal $t \in T' \setminus \{s\}$ on a graph with nonnegative edge costs π_{ij}^{st}. Hence, the pricing subproblem of UPath can be solved in polynomial time as well. For UPath$^{\mathrm{B}}$, we need to additionally consider the dual variable values of constraints (14), while for UPath$^{\mathrm{D}}$ and UPath$^{\mathrm{DI}}$ we replace variables π_{ij}^{st} by the dual variables of constraints (17) defined on the directed arc set A. In both cases, however, the general structure of the pricing subproblem remains identical, i.e. we need to solve HCSPPs between root and terminal nodes on a graph with nonnegative arc costs.

Following an approach proposed by Gouveia et al. [9] for the distance constrained minimum spanning tree problem, we add multiple path variables for each root terminal pair when solving the pricing subproblem within branch-and-price. Here, we first solve the HCSPP for the current root $s \in R$ and terminal $t \in T'$, $s \neq t$, and then consider all nodes $i \in V$ adjacent to terminal t and each hop value $h = 0, \ldots, H-1$. If a path g from s to i with h hops has been computed, that is if g is cheaper than all paths from s to i with less than h hops, and $p = g \cup \{(i,t)\}$ yields negative reduced costs, p is added to the restricted master problem. A set of initial paths is generated using the same procedure but original edge costs instead dual variable values. In case this strategy does not yield a feasible LP, artificial path variables (with very large cost) each consisting of an empty edge set are added between each root and relevant terminal.

Branching. We apply branching on fractional node, edge, and disaggregated arc variables, respectively, in our branch-and-price approaches, since this simple branching rules do not alter the structure of the pricing subproblem.

5 Computational Results

We used benchmark instance sets TR and TC from [8], each consisting of five complete graphs containing 31 or 41 nodes and either random (TR) or euclidean (TC) edge costs, respectively. For each instance, the first $|T|$ nodes are used as terminals and the last $|R|$ nodes as root nodes. Furthermore, we used sparse instances from the OR-Library originally proposed for the Steiner tree problem in graphs [4]. Six concrete instances from class B that turned out to be nontrivial and to allow for feasible solutions in a reasonable range of H have been chosen. For these instances a subset of nodes is already defined as terminals. Among the latter, we choose the first $|R|$ to be roots ($|R| \in \{2, 4\}$), while the remaining original terminals define the set of terminals of the resulting instance of the HCSTPMR.

Our computational study has been performed on a multi-core system where each eight cores share 24GB RAM. Each run has been performed on a single core of an Intel Xeon E5540 processor with 2.53 GHz using a C++ implementation based on SCIP 2.0.2 [1] with CPLEX 12.2 as embedded LP solver. When solving path models by branch-and-price, we configured SCIP to use the dual simplex as LP solver since this option outperformed alternative approaches. Parameter "fastmip" has been set to one and presolving has been disabled for all branch-and-price approaches. We applied a memory limit of 4GB to each individual experiment and an absolute time limit of 3 600 CPU-seconds for solving LP-relaxations and of 7 200 CPU-seconds for solving the MIP models, respectively. Apart from that, default settings and plugins of SCIP have been used.

Figure 3 summarizes our computational results on instance sets TC and TR. We conclude that, in particular for TR instances, the LP bounds of the path models are significantly tighter than the ones of the flow models. These results are consistent with the results given in [8] for the single root case. Since using bidirectional commodity-pair forcing constraints yields higher CPU-times but worse LP bounds than the disaggregated models with node-degree constraints, we did not consider the former when solving the integer models. We note that for these instances, solving the LP relaxations of the path models usually takes longer than solving the one of the weaker flow models. Regarding total CPU-times for solving the integer models as shown in Figure 3e and 3f, we conclude that this additional effort clearly pays off if the hop limit is not too loose or if the difference in terms of LP bounds is not too small, i.e. the path models outperform the flow models for all cases except for instances TC with $H = 5$. Notice that for some settings, the LP relaxations could not be solved within the given time limit. This explains, smaller average relative LP bound improvements of UFlow^{B} than of UFlow in Figures 3c and 3d, respectively.

Results for instances from the OR-Library are summarized in Table 1. Here we directly use the compact flow models instead of applying branch-and-price based on edge variables since the overhead of the latter did not pay off for these sparse instances. We observe that UFlow and UFlow^{DI} usually exhibit a significant gap between LP and MIP objective value, while in particular UPath^{DI} successfully closes this gap for the majority of test cases. We conclude that UPath^{DI} exhibits

Table 1. Relative difference between LP bound $v_{\mathrm{LP}}(.)$ and optimal IP value *opt* in % and CPU-times in seconds for solving MIP models on Steinlib instances

						$(opt - v_{\mathrm{LP}}(.))/opt$ [%]				CPU-time [s]			
Inst.	$\|V\|$	$\|E\|$	$\|R\|$	$\|T\|$	H	UFlow	UFlow$^{\mathrm{DI}}$	UPath	UPath$^{\mathrm{DI}}$	UFlow	UFlow$^{\mathrm{DI}}$	UPath	UPath$^{\mathrm{DI}}$
B10	75	150	2	11	5	4.26	2.96	0.38	**0.00**	32	59	**0**	**0**
B10	75	150	2	11	6	4.25	3.47	**0.00**	**0.00**	13	11	**0**	1
B10	75	150	2	11	7	1.85	1.41	**0.00**	**0.00**	11	6	**0**	1
B10	75	150	2	11	8	1.93	1.35	0.86	**0.00**	18	3	**2**	**2**
B10	75	150	2	11	9	1.30	0.91	0.57	**0.00**	**1**	4	2	2
B10	75	150	4	9	6	14.27	12.73	3.48	**0.00**	148	1291	**6**	12
B10	75	150	4	9	7	1.33	0.99	**0.00**	**0.00**	23	11	**2**	13
B10	75	150	4	9	8	2.44	2.22	1.11	**0.00**	44	31	**5**	22
B10	75	150	4	9	9	1.21	0.91	**0.00**	**0.00**	**3**	13	5	23
B11	75	150	2	17	5	18.20	13.06	7.49	**0.34**	4128	1541	26	**4**
B11	75	150	2	17	6	19.77	16.18	10.64	**4.55**	2318	2523	307	**65**
B11	75	150	2	17	7	6.01	2.76	0.44	**0.00**	106	87	10	**8**
B11	75	150	2	17	8	3.28	0.82	0.57	**0.00**	98	28	**12**	21
B11	75	150	2	17	9	1.94	**0.00**	0.28	**0.00**	41	**7**	12	23
B11	75	150	4	15	6	24.64	20.63	13.76	**6.05**	7200	7200	1395	**563**
B11	75	150	4	15	7	5.99	2.76	0.35	**0.00**	259	859	**27**	112
B11	75	150	4	15	8	3.28	0.82	0.55	**0.00**	344	222	**28**	106
B11	75	150	4	15	9	1.84	**0.00**	0.28	**0.00**	89	**20**	34	153
B12	75	150	2	36	6	9.93	8.37	6.14	**2.48**	7200	7200	2649	**193**
B12	75	150	2	36	7	4.39	3.15	1.60	**0.00**	2068	3256	71	**22**
B12	75	150	2	36	8	2.47	1.45	0.97	**0.00**	1057	852	74	**36**
B12	75	150	2	36	9	2.53	1.80	1.46	**0.00**	1808	2446	591	**69**
B12	75	150	4	34	6	9.83	7.28	4.10	**0.00**	7200	7200	345	**124**
B12	75	150	4	34	7	5.23	3.44	2.25	**0.00**	7200	7200	310	**142**
B12	75	150	4	34	8	4.11	2.86	2.39	**0.00**	7200	7200	582	**511**
B12	75	150	4	34	9	6.64	5.68	5.19	**3.39**	7200	7200	7200	**1740**
B16	100	200	2	15	7	10.18	7.76	4.50	**0.00**	1087	1461	34	**9**
B16	100	200	2	15	8	4.58	2.49	**0.00**	**0.00**	111	105	**6**	16
B16	100	200	2	15	9	4.54	3.32	0.03	**0.00**	181	74	**18**	28
B16	100	200	4	13	7	7.78	5.60	3.33	**0.00**	1192	7200	152	**86**
B16	100	200	4	13	8	4.48	2.41	**0.00**	**0.00**	311	322	**31**	200
B16	100	200	4	13	9	4.49	3.30	**0.00**	**0.00**	599	307	**56**	290
B17	100	200	2	23	6	6.50	4.15	2.67	**0.00**	292	568	5	**1**
B17	100	200	2	23	7	5.79	2.66	1.77	**0.00**	368	240	**4**	5
B17	100	200	2	23	8	5.43	1.84	2.94	**0.00**	317	332	15	**8**
B17	100	200	2	23	9	4.39	1.18	2.63	**0.00**	227	40	**15**	16
B17	100	200	4	21	8	9.47	4.56	6.67	**0.89**	7200	7200	**1589**	3245
B17	100	200	4	21	9	7.35	3.46	4.75	**0.00**	7200	7200	693	**582**
B18	100	200	2	48	6	5.13	4.08	2.43	**0.00**	7200	7200	196	**29**
B18	100	200	2	48	7	4.45	3.29	1.36	**0.00**	7200	7200	432	**57**
B18	100	200	2	48	8	2.23	0.61	0.48	**0.00**	1969	472	101	**89**
B18	100	200	2	48	9	2.80	1.09	1.56	**0.22**	6736	2292	1001	**328**

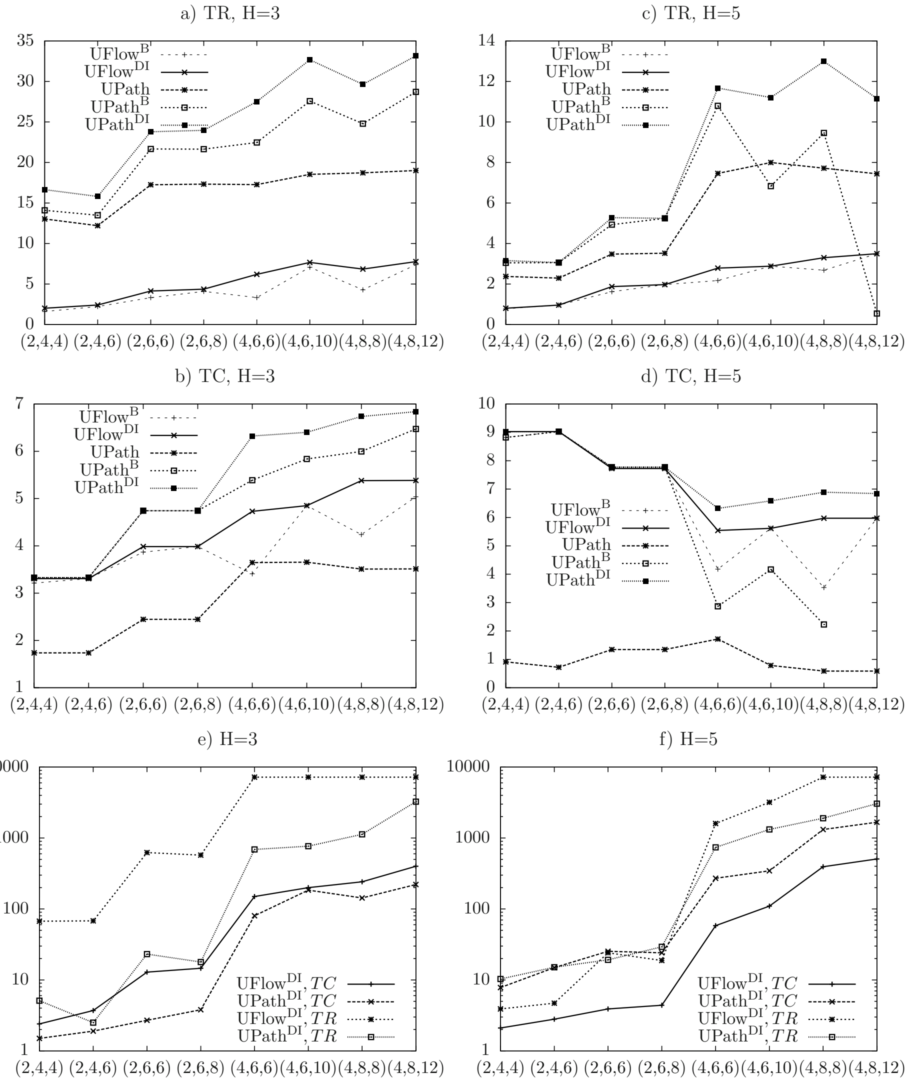

Fig. 3. a),b),c),d) Average relative improvement of LP bound v_{LP} over UFlow, i.e. $(v_{\text{LP}}(.) - v_{\text{LP}}(\text{UFlow}))/v_{\text{LP}}(\text{UFlow})$, in % and e), f) median CPU-times in seconds (MIP) for various values of $(|R|, |T|, |T'|)$

the best overall performance and outperforms the other options proposed in this paper. Moreover, both path models significantly outperform the flow models on sparse instances, while the observed difference in their performance was considerably smaller on the complete instance sets.

6 Conclusion

In this paper, we introduced a generalization of the hop constrained minimum Steiner tree problem on a graph involving multiple root nodes. Since, due to multiple hop-constraints, it is not straightforward to orient feasible solutions, we introduced undirected flow and path MIP formulations which have been further strengthened using bidirectional commodity-pair forcing constraints and disaggregation of design variables. We further proposed branch-and-price approaches for our models. Computational results show that the branch-and-price approaches based on path formulations outperform the flow formulations if the hop limit is not too loose. Furthermore, the relative performance difference between path and flow models significantly increases when the instance graphs are sparse. Our future work on the HCSTPMR includes studying the possibilities to model the problem over layered graphs and the development of corresponding branch-and-cut approaches.

References

1. Achterberg, T.: SCIP: Solving constraint integer programs. Mathematical Programming Computation 1(1), 1–41 (2009)
2. Balakrishnan, A., Altinkemer, K.: Using a hop-constrained model to generate alternative communication network design. INFORMS Journal on Computing 4(2), 192–205 (1992)
3. Balakrishnan, A., Magnanti, T.L., Mirchandani, P.: Modeling and heuristic worst-case performance analysis of the two-level network design problem. Management Science 40(7), 846–867 (1994)
4. Beasley, J.E.: Collection of test data sets for a variety of operations research (OR) problems (1990), `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`
5. Botton, Q., Fortz, B., Gouveia, L., Poss, M.: Benders decomposition for the hop-constrained survivable network design problem. INFORMS Journal on Computing (2011) (to appear)
6. Dahl, G., Gouveia, L., Requejo, C.: On formulations and methods for the hop-constrained minimum spanning tree problem. In: Resende, M.G.C., Pardalos, P.M. (eds.) Handbook of Optimization in Telecommunications, pp. 493–515. Springer (2006)
7. Frangioni, A., Gendron, B.: A stabilized structured Dantzig-Wolfe decomposition method. Mathematical Programming (to appear, 2012)
8. Gouveia, L.: Multicommodity flow models for spanning trees with hop constraints. European Journal of Operational Research 95(1), 178–190 (1996)
9. Gouveia, L., Paias, A., Sharma, D.: Modeling and solving the rooted distance-constrained minimum spanning tree problem. Computers & Operations Research 35(2), 600–613 (2008)
10. Gouveia, L., Simonetti, L., Uchoa, E.: Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. Mathematical Programming 128, 123–148 (2011)
11. Gouveia, L., Telhada, J.: The multi-weighted Steiner tree problem: A reformulation by intersection. Computers & Operations Research 35, 3599–3611 (2008)
12. Ljubić, I.: Exact and Memetic Algorithms for Two Network Design Problems. PhD thesis, Vienna University of Technology (2004)