A New Layered Graph Approach to Hop- and Diameter-constrained Spanning/Steiner Tree Problems in Graphs

Markus Sinnl and Ivana Ljubić

Department of Statistics and Operations Research,
University of Vienna, Vienna, Austria
{markus.sinnl,ivana.ljubic}@univie.ac.at

Abstract. In this paper a novel generic way to model hop- or diameter-constrained tree problems as integer linear programs is introduced. The concept of layered graphs has gained widespread attention in the last few years, since it exhibits significant computational advantages when compared to previously available extended, but compact, formulations for this type of problems. We derive a new extended formulation on a layered graph in which the underlying optimization problem can be modeled as the Steiner arborescence problem with additional degree constraints. The power of our new model is demonstrated on the Steiner Tree Problem with Revenues, Budget and Hop Constraints, for which a branch-and-cut algorithm has been implemented. Most of the instances available for the DIMACS-challenge, including many previously unsolved ones, can be solved to proven optimality within a time limit of 1000 seconds, most of them within a few seconds only.

1 Introduction

In this work, we present a novel generic way to model hop- or diameter-constrained tree problems as integer linear programs (ILPs). In this type of problems we typically search for a subtree of a given input graph G such that from a given root node r to any other node of this subgraph, there exists a path containing at most H edges (where $H \geq 2$ is a given hop-limit). Our approach is based on layered graphs, a concept which has gained widespread attention in the last few years. On the one hand, layered graphs allow for significant improvements of computing times when compared to previously available extended formulations (see, [1]). On the other hand, they are also shown to theoretically dominate most of the available extended formulations that model hop- or diameter-constraints. Instead of modeling the problem on G, a layered graph is constructed such that for each layer $1 \leq h \leq H$, a copy of the nodes of G is established, and nodes between two consecutive layers are connected whenever there exists a connection between them in G (for more details, see Section 2). In the approaches from the available literature, the underlying problem is then formulated as a Steiner arborescence problem using arc variables on such obtained layered digraph. While these formulations often provide very good LP-bounds (see, e.g. [1]), the number of variables (which is $O(H|E|)$), often becomes prohibitive when the problem is formulated on larger graphs, or when larger hop limits H are considered. Instead, our new formulation comprises only node variables on the layered graph (along with node and arc variables on G). Whereas standard layered graph approaces involve $O(H|V|^2)$ variables, our new model deals with $O(H|V| + |E|)$ variables.

We demonstrate the power of our new modeling approach on the *Steiner Tree Problem with Revenues, Budget and Hop Constraints (STPRBH)* which has been taken as part of the DIMACS Challenge. A branch-and-cut-algorithm derived on our new formulation solves most of the instances from the DIMACS Challenge to provable optimality in a short time (within a few seconds). This includes many instances for which the optimal solution has been unknown. We also provide results for the *Hop Constrained Spanning Tree Problem (HCSpT)* studied by [1].

Our paper is structured as follows: In the remainder of this section, the problem definition of the STPRBH is given, together with a short literature overview. In Section 2, a short review of layered graphs is followed by the presentation of our generic new model together with some improvements. The proposed improvements include strengthening of valid inequalities, fixing/removing of variables and introduction of further valid inequalities. We also demonstrate how to formulate the STPRBH and the HCSpT with our new model.

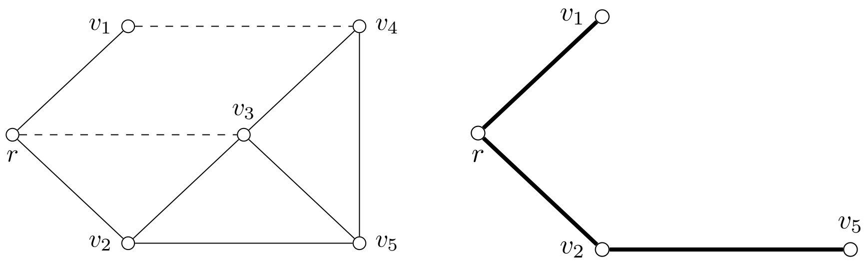
Section 3 contains a description of our algorithmic framework together with obtained computational results. Section 4 concludes the work with a short summary and a discussion of future work. It points out a broader potential of the proposed model.

It should be noted that this paper presents some preliminary results of an ongoing study and further improvements of the presented results can be expected.

Definition 1 (Steiner Tree Problem with Revenues, Budget and Hop Constraints (STPRBH)). *We are given an undirected graph $G = (V, E)$ with edge costs $c : E \mapsto \mathbb{R}^+$, node revenues $p : V \mapsto \mathbb{R}^+$ and a dedicated root node $r \in V$, a hop limit $H \in \mathbb{N}^+$ and a budget limit $B \in \mathbb{R}^+$.*

A feasible solution of the STPRBH is a subtree $\mathcal{T} = (V_S \subseteq V, E_S \subseteq E)$ rooted at r, where every node in V_S can be reached form the root r using at most H edges and the total cost of the edges in E_S does not exceed B, i.e., $\sum_{e \in E_S} c_e \leq B$. The goal is to find a feasible subtree $\mathcal{T}^*$ that maximizes the revenue defined as $\sum_{v \in V_S} p_v$.

The problem has been introduced in [2] where three branch-and-cut approaches have been presented: one based on Miller-Tucker-Zemlin constraints, one on Dantzig-Fulkerson-Johnson (also known as subtour-elimination) constraints, and one on hop-indexed formulation. Note that the latter formulation is based on hop-indexed edge variables, i.e., it can be viewed as an edge-based layered graph approach. Instances derived from sets B and C of the OR-library [3] have also been introduced in [2]. All instances from the set B and instances $C1$ to $C5$ have been solved to optimality with the approaches from [2]. However,



(a) Instance. (b) Solution.

Fig. 1: (1a) Graph of an instance of the STPRBH problem. Let $p_1 = 10, p_2 = 0$, $p_3 = 4, p_4 = 9, p_5 = 5$, the cost of the solid edges be one, and of the dashed edges be five. (1b) The optimal solution for $H = 2$ and $B = 3$ is and has objective value 15.

no single model works well for all instances. In [4], the same authors proposed a greedy heuristic and a tabu search with some improvement procedures. They also report some results for $C6$ to $C20$. According to [4], for these instances, not even the root relaxation could be solved within a time limit of two hours (in most of the cases). Branch-and-price approaches for the STPRBH have been studied by Sinnl [5], and in [6] a lifted Miller-Tucker-Zemlin formulation and a formulation based on reformulation-linearization techniques were given. These two latter papers report on computational results on the instances from set B and $C1$ to $C5$, but offer no consistent speed up, when compared to [4]. Recently, a breakout local search algorithm (see [7]) and a memetic algorithm (see [8]) have been proposed. These two recent papers provide improved feasible solutions for some of the unsolved instances ($C6$ to $C20$). Some new instances based on graphs $C16$ to $C20$ are also introduced in [8].

2 Problem Formulation and Enhancements

Let $G_L = (V_L, A_L)$ be the layered graph associated with a rooted graph $G(V, E)$ and hop limit H. It can be defined as follows (see, e.g., [1]): The node set $V_L = r \cup V^1 \cup V^2 \cup \ldots \cup V^H$, where V^h contains a copy v^h of a node $v \in V$, iff v can be reached by a path of exactly h edges in G. Note that the root node r is the only node on layer zero. The arc set $A_L = A^1 \cup A^2 \cup \ldots \cup A^H$, where A^h contains a directed copy (i, j) of an edge $\{i, j\} \in E$, iff $i \in V^{h-1}$ and $j \in V^h$. Thus the layered graph has size $O(H(|V| + |E|))$. Figure 2 shows the layered graph associated with our exemplary instance from Figure 1a and $H = 3$.

In previous approaches from literature, hop-constrained problems are formulated on G_L by associating variables to the arcs A_L of the layered graph, e.g., x^h_{ij} is one, if arc $\{i, j\}$ is used on layer h. While this usually gives models with strong LP-bounds, the size of the resulting models soon becomes prohibitive. We

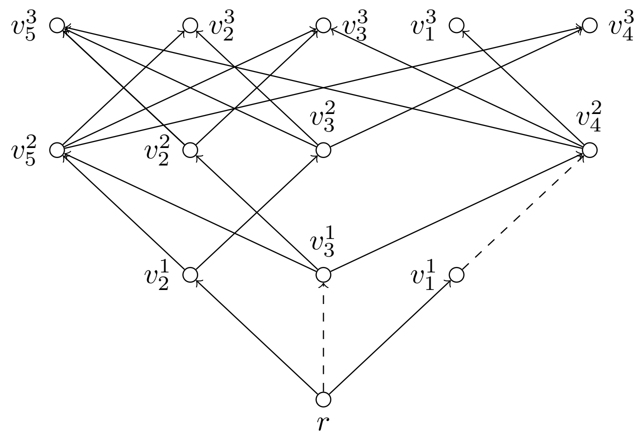


Fig. 2: Layered graph associated with the graph from (1a) and $H = 3$.

thus propose to project out arc variables from the layered graph and model the hop-constraints by associating variables with the nodes V_L of the layered graph.

To do so, we transform the graph G into a rooted digraph $D = (V, A)$, where A are the bidirected edges from E. We use the following sets of binary variables to model our problem:

$$x_a = \begin{cases} 1 & \text{if arc } a \text{ is part of the solution} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } a \in A;$$

$$y_v = \begin{cases} 1 & \text{if node } v \text{ is part of the solution} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } v \in V;$$

Variables x and y are used to model a rooted arborescence in G (similar to the standard approaches to the prize-collecting Steiner trees). Additional node variables y_v^h are introduced to model the distance of the node v from the root r, i.e.:

$$y_v^h = \begin{cases} 1 & \text{if node } v \text{ is on layer } h \text{ in the solution} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } v \in V, h \in H.$$

The y^h-variables, together with the x-variables are used to ensure that the solution satisfies the hop constraint.

Let $\delta^-(W) = \{(i,j) \in A : i \notin W, j \in W\}$ and $\delta^+(W) = \{(i,j) \in A : i \in W, j \notin W\}$. Let T denote the set of all terminals (i.e., depending on the problem, nodes, which must be in any feasible solution, or nodes with positive revenue) and let $S = V \setminus T$. Using this notation, we obtain a generic model (NODEHOP) for hop-constrained tree problems:

$$
\begin{aligned}
&\text{(NODEHOP)} \quad & x(\delta^-(W)) &\geq y_v & \forall W \subseteq V, v \in W \cap T, r \notin W \quad & (1)\\
&& y_r &= 1 & & (2)\\
&& x(\delta^-(v)) &= y_v & \forall v \in V \quad & (3)\\
&& \sum_{h \in H} y_v^h &= y_v & \forall v \in V \quad & (4)\\
&& x_{rv} &= y_v^1 & \forall (r,v) \in A \quad & (5)\\
&& y_v^{h-1} + x_{vw} &\leq 1 + y_w^h & \forall (v,w) \in A, v \neq r, 2 \leq h \leq H \quad & (6)\\
&& y_v^H + x_{vw} &\leq 1 & \forall (v,w) \in A, v \neq r \quad & (7)\\
&& x_a &\in \{0,1\} & \forall a \in A \quad & (8)\\
&& y_v &\in \{0,1\} & \forall v \in V \quad & (9)\\
&& y_v^h &\in \{0,1\} & \forall v \in V, h \in H \quad & (10)
\end{aligned}
$$

Constraints (1), (2) and (3) is the cut formulation for the (prize-collecting) Steiner tree problem (see, e.g. [9]) and ensures that our solution is an arborescence rooted at r.

The remaining set of inequalities (4)-(7) deals with the hop constraint: If a node is part of the solution, it must lie on some layer (4) and if a node lies on the last layer which is feasible, i.e., H, there can be no outgoing arc from it (7). Moreover, if the arc going from the root to node v is used, node v must lie on layer 1, this is ensured by (5). Constraints (6) make sure that if a node v lies on layer $h-1$ ($2 \leq h \leq H$) and arc (v, w) is taken in the solution, then node w must lie on layer $h+1$. Note that crucial for the validity of our model is the tree/arborescence property: since every node only has one incoming arc (see constraints (3)), the layer of each node is uniquely defined. Thus, inequalities (1) to (7) ensure in a generic way that the solution is an arborescence, satisfying the hop-constraint.

2.1 Modeling the STPRBH and the HCSpT

Using the generic model NODEHOP, it is easy to obtain the following formulation for the STPRBH:

$$
\begin{aligned}
&\text{(STPRBH)} \quad & \max \sum_{v \in T} p_v y_v & \quad (11)\\
&& \sum_{a \in A} c_a x_a \leq B & \quad (12)\\
&& (x, y, y^h) \in NODEHOP & \quad (13)
\end{aligned}
$$

The objective function (11) ensures maximization of the profit, while constraint (12) makes sure that a solution does not exceed the given budget B. Similarly, we can model the HCSpT as follows:

$$\text{(HCSpT)} \qquad \min \sum_{a \in A} c_a x_a \tag{14}$$

$$y_i = 1 \qquad \forall i \in V \tag{15}$$

$$(x, y, y^h) \in NODEHOP \tag{16}$$

The objective function (14) ensures minimization of the cost, and constraints (15) ensure that all nodes are in the solution. Clearly, by specifying constraints (15) for a given terminal set $T \subset V$, instead of all V, the *Hop Constrained Steiner Tree Problem* can also be modeled in our framework.

2.2 Improving the Model

In the following we provide some improvements of the proposed model. Some of these improvements are only possible, when $T \neq V$, i.e., for Steiner Tree problems, but not for their spanning tree counterparts.

Fixing of Variables Obviously, no node v, where $(r, v) \notin A$ can lie on the first layer, thus also all variables corresponding to such nodes can be fixed to zero. This fixing can be enhanced with the help of a breadth-first-search (BFS) that calculates the shortest distance (in terms of the number of edges) between the two nodes u and v, denoted by $dist(u, v)$. Let BFS_r be the tree resulting from a BFS starting at a root node, then $dist(r, v)$ denotes also the layer on which node v lies in this tree. Consequently, all variables y_v^h with $h < dist(r, v)$ can clearly be fixed to zero. For a similar approach, see also [10].
Moreover, it is easy to see that there always exists an optimal solution for STPRBH (and other Steiner tree problems), where no Steiner node S is a leaf. It follows that no node in S can lie on the last layer H, thus we can fix all y_v^H, $v \in V \setminus T$ to zero.
Finally, for a node $v \in V$, let $dist(v, T) = \min_{w \in T} dist(v, w)$ be the distance between v and a closest node from the terminal set T. By definition, if $v \in S$, we must cross at least $dist(v, T) - 1$ layers in order to reach a node in T from v. It follows that all variables y_v^h with $h > H - dist(v, T)$ can be fixed to zero. Consequently, all variables y_v, with $dist(r, v) + dist(v, T) > H$, can be fixed to zero as well (resp. removed in a preprocessing step).

Valid Inequalities

Hop-Link and Hop-Link-End Inequalities First, note that in both constraints (6) and (7), the value 1 can be lifted down to y_v. The constraints still remain valid, since any of y_v^{h-1}, y_v^H and x_{vw} set to one also implies that y_v is set to one. The obtained constraints are

$$y_v^{h-1} + x_{vw} \leq y_v + y_w^h, \quad \forall (v, w) \in A, v \neq r, 2 \leq h \leq H \tag{HLink}$$

They will be called *hop-link constraints*, as they are linking nodes of two consecutive layers with an arc. Similarly, for the nodes at the last layer, we will call the following constraints, *hop-link-end constraints*:

$$y_v^H + x_{vw} \leq y_v, \quad \forall (v, w) \in A, v \neq r \tag{HLink$_e$}$$

Lifted Hop-Link and Hop-Link-End Inequalities These two inequalities can further be improved by observing that an arc (v, w) must start at some layer $k \leq H - dist(w, T) - 1$, because node w must lie in a layer $\leq H - dist(w, T)$ in an optimal solution. Let $k^* = \max\{h, H - dist(w, T)\}$. We can add $\sum_{k \geq k^*} y_i^k$ to the left-hand-side of (HLink)

Theorem 1. *Let $(v, w) \in A$, $v \neq r$, $2 \leq h \leq H$ and $k^* = \max\{h, H - dist(w, T)\}$. Then the* lifted hop-link *inequality*

$$y_v^{h-1} + \sum_{k=k^*}^{H} y_v^k + x_{vw} \leq y_v + y_w^h \tag{l-HLink}$$

is valid for NODEHOP.

Proof. If all y_v^k, $k \geq k^*$ are zero, the inequality reduces to (HLink). Thus, suppose some y_v^k is one. Due to the fixing of variables y_w^k to zero (for $k \geq H - dist(w, T)$) and inequalities (HLink), (HLink$_e$), it follows that $x_{vw} = 0$, when some y_v^k is one. Since the sum starts at $\max\{h, H - dist(w, T)\}$, every y_v^h only appears at most once on the left-hand-side and due to equalities (4), the left-hand-side in this case is one and the right-hand-side is at least one. □

We distinguish the following two cases:

- $w \in S$: Notice that for $h \geq H - dist(w, T)$ this inequality boils down into:

  $$\sum_{k=h-1}^{H} y_v^k + x_{vw} \leq y_v, \quad \forall (v, w) \in A, v \neq r, H - dist(w, T) \leq h \leq H, w \in S \tag{17}$$

  since y_w^h is fixed to zero. One easily observes than all inequalities of type (17) for $h > H - dist(w, T)$ are dominated by the single inequality of the same type for $h = H - dist(w, T)$. This also holds for (HLink$_e$), which is actually (17) for $h = H$.
- $w \in T$: Similarly, for $w \in T$, inequality (l-HLink) becomes:

  $$y_v^{h-1} + y_v^H + x_{vw} \leq y_v + y_w^h, \quad \forall (v, w) \in A, v \neq r, 2 \leq h \leq H, w \in T \tag{18}$$

  Observe that (18) is just the inequality (HLink) lifted from the left-hand-side by y_v^H.

Generalized Hop-Link and Hop-Link-End Inequalities Using constraint (4) corresponding to node v, inequality (HLink) for an arc (v, w), $v \neq r$ and a given layer $h : 2 \leq h \leq H$ can be rewritten as

$$x_{vw} \leq \sum_{k \in H_1, k \neq (h-1)} y_v^k + y_w^h \tag{19}$$

where $H_1 := \{1, \ldots, H\}$. It has the intuitive meaning that if arc (v, w) is in the solution, it either ends at layer h (and thus has started at layer $h-1$), or it must have started at some other layer than $h-1$. Consider now another layer $l \neq h$: Inequality (19) is valid, because when arc (v, w) ends at layer l, it must have started at layer $l-1$ and there is y_v^{l-1} on the right-hand-side of (19).

To motivate the generalization of these inequalities, observe that when the arc (v, w) ends at some layer $\neq l, h$, the variable y_v^{l-1} must be zero in a valid solution. Moreover, when arc (v, w) ends at layer l, the variable y_w^l must be one in any feasible solution. Thus it follows that y_v^{l-1} can be replaced by y_w^l in constraint (19) and the constraint remains valid. Generalizing this idea further, we observe that for each layer $h \geq 2$, in the summation on the right-hand-side, we must either include y_v^{h-1} or y_w^h. This brings us to the following family of inequalities:

Theorem 2. *Let $H_2 = \{2, \ldots, H\}$ and P be the family of functions $P = 2^{H_2}$, and $(v, w) \in A, v \neq r$. Then the* generalized hop-link *inequality*

$$x_{vw} \leq \sum_{h \in H_2} \left(p_h y_v^{h-1} + (1 - p_h) y_w^h\right) \tag{g-HLink}$$

is valid for NODEHOP.

Proof. Clearly, when node w lies on layer 1 it must be connected to the root node and x_{vw} must be zero. Thus suppose there exists a feasible solution, where node w lies on some layer $k : 2 \leq k \leq H$, x_{vw} is one, i.e., the arc (v, w) is used and the right-hand-side of (g-HLink) is zero. Since node w lies on layer k and the arc (v, w) is used, it follows that node v must lie on layer $k-1$. This implies that both y_v^{k-1} and y_w^k are one. Due to the definition of the function p, $p_k = 1$ or $p_k = 0$ and consequently, we have either y_v^{k-1} or y_w^k on the right-hand-side and thus the right-hand-side is one, which is a contradiction to the assumption that the inequality is violated. $\square$

For each arc $(v, w) \in A$, constraints (g-HLink) can easily be separated in $O(H)$ time: Given a fractional solution $(\tilde{x}, \tilde{y}, \tilde{y}^h)$, for each layer $h \geq 2$, we consider the sum $\sum_{h \in H_2} \min\{\tilde{y}_v^{h-1}, \tilde{y}_w^h\}$. If the obtained sum is smaller than $\tilde{x}_{vw}$, a violated constraint is detected.

Let us now consider the special mapping $p \in P$, such that $p_h = \begin{cases} 1 & \text{if } h \text{ is even} \\ 0 & \text{otherwise} \end{cases}$.

Then, (g-HLink) becomes:

$$x_{vw} \leq \begin{cases} \sum_{h \in H_2, h \text{ odd}} \left(y_v^h + y_w^h\right) + y_v^1 - y_v^H, & H \text{ odd} \\ \sum_{h \in H_2, h \text{ odd}} \left(y_v^h + y_w^h\right) + y_v^1, & \text{otherwise} \end{cases} \tag{20}$$

Observe that we need the case distinction due to the range in the summation, in case that H is odd, we have y_v^H in the sum, which is not implied by inequalities (g-HLink) and we thus subtract it again in the end. For ease of notation, we assume that H is odd in the following, the case with H being even works analogously. By summing-up inequalities (20) associated to arcs (v, w) and (w, v), we end up with

$$x_{vw} + x_{wv} \leq \sum_{h \in H_2, h \text{ odd}} 2\big(y_v^h + y_w^h\big) + y_v^1 + y_w^1 - y_v^H - y_w^H \tag{21}$$

After rewriting the right-hand-side as $\sum_{h=2,h \text{ odd}}^{H-1} 2\big(y_v^h + y_w^h\big) + y_v^1 + y_w^1 + y_v^H + y_w^H$, we can down-lift the coefficient 2 on the right-hand-side to 1 (since $x_{vw} + x_{wv} \leq 1$). Thus, the validity of the new derived inequalities presented in the following theorem follows immediately.

Theorem 3. *Let $(v, w) \in A$, $v \neq r$. Then the* odd two-arc hop-link *inequality*

$$x_{vw} + x_{wv} \leq \sum_{h \in H_1, h odd} \big(y_v^h + y_w^h\big) \tag{o2AHLink}$$

is valid for NODEHOP.

Starting now with the mapping $p \in P$, such that $p_h = \begin{cases} 0 & \text{if } h \text{ is even} \\ 1 & \text{otherwise} \end{cases}$ and using similar arguments, we end up with the following family of valid inequalities.

Theorem 4. *Let $(v, w) \in A$, $v \neq r$. Then the* even two-arc hop-link *inequality*

$$x_{vw} + x_{wv} \leq \sum_{h \in H_2, h even} \big(y_v^h + y_w^h\big) \tag{e2AHLink}$$

is valid for NODEHOP.

Cut Inequalities on the Layered Graph If a node w lays on a layer h, there obviously must be at least one node $v \neq w$ at layer $h - 1$ in the solution. This leads to the following family of *node-hop-index* inequalities:

$$\sum_{(v,w) \in A} y_v^{h-1} \geq y_w^h \tag{22}$$

Such inequalities (expressed in terms of arc-variables on the layered graph) are commonly used in the hop-indexed models for hop-constrained problems (see, e.g. [11]). They represent a compact way of ensuring a connectivity of a solution. However, these hop-indexed compact models are known to suffer from weak lower bounds. In state-of-the-art approaches, connectivity constraints are therefore modeled using cut-set inequalities on layered graphs (see, e.g. [1,10]). In a similar fashion, we are currently working on a generalization of cut-set inequalities on the layered graph using y^h and x variables only.

Hereby, we illustrate a subfamily of desired cut-set inequalities that is used in our current computations. Observe first that if the input graph is complete, node-hop-index inequalities will be in general very weak, since the left-hand-side contains all nodes on layer $(h-1)$ in this case. Clearly, also the following inequality holds for any h and node $w \neq r$, since it is a weaker version of inequalities (1) for $W = \{v\}$:

$$\sum_{(v,w)\in A} x_{vw} \geq y_w^h \tag{23}$$

Observe that in both (22) and (23), the right-hand-side is the same, and we sum over all arcs on the left-hand-side. Hence, we can derive a more general family of inequalities, which contains both (22) and (23) as a special case.

Theorem 5. *Let R be the family of functions $R = 2^A$ and $r \in R$, $w \in V$ and $2 \leq h \leq H$. Then the* node-arc-cut-inequalities

$$\sum_{(v,w)\in A} \Big(r_{vw} x_{vw} + (1 - r_{vw}) y_v^{h-1} \Big) \geq y_w^h \tag{NACut}$$

are valid for NODEHOP.

Proof. Suppose there exists a feasible solution, where y_w^h is one, i.e., node w lies on layer h, and the left-hand-side is zero. However, since the node lies on layer h, there must be an incoming arc (v,w) from some node v lying on layer $h-1$, thus both y_v^{h-1} and (v,w) must be one. One of these variables is on the left-hand-side of constraint (NACut), and thus the left-hand-side is one, which concludes the proof. $\square$

Constraints (NACut) can be separated in polynomial time as follows: Given a fractional solution $(\tilde{x}, \tilde{y}, \tilde{y}^h)$ and a node w and layer h, consider all nodes v, such that $(v,w) \in A$, and calculate the sum $\sum_{v:(v,w)\in A} \min\{\tilde{x}_{vw}, \tilde{y}_v^{h-1}\}$. If the resulting sum is smaller than the LP-value of y_w^h, a violated inequality is obtained.

3 Computational Results

We have implemented branch-and-cut algorithms for the STPRBH and the HC-SpT based on our model. The computational results are obtained using a single core of an Intel E5-2670v2 with 2.5GHz and 64GB RAM and CPLEX 12.6 as ILP-solver. The following general purpose cuts of CPLEX have been set to one (moderate generation of cuts): `fractional`, `zero-half`, `cover`, all the other cuts are left at the default parameter.

Our initial model for both the STPRBH and the HCSpT consists of (2), (3), (4),(5), (l-HLink) for $h = k^* - 1$, i.e., the lifted version of (HLink$_e$) and (e2AHLink). Moreover, inequalities

$$x_{ij} + x_{ji} \leq y_i, \quad i, j \in V,$$

which are (1) for $|W| = 2$ are added. Inequalities (1), as well as inequalities (l-HLink) and (NACut) are separated "on the fly". Constraints (1) are separated using a max-flow separation, when the LP-solution is fractional (see, e.g., [9], [12] for details), and with a breadth-first search, when the LP-solution is integer. Constraints (l-HLink) are of polynomial size and are separated by enumeration and (NACut) are separated as described above. Depending on the problem, additional constraints have been used, these constraints are mentioned in the respective sections for the STPRBH and the HCSpT. The above model is called IMPROVED. In the following, we also report results for a basic model, where the valid inequalities (e2AHLink) and (NACut) are not used, this model is denoted by BASIC.

Our algorithm also contains a primal heuristic which is called after each LP at the root node and at the end of each node in the branch-and-bound tree. Moreover, for the STPRBH, we explicitly turned on the CPLEX heuristics, while for HCSpT, we left it at the default setting, since this has proven advantageous in our initial testruns.

3.1 STPRBH

The initial model additionally contains inequality (12) and the flow-balance inequalities

$$x(\delta^-(v)) \leq x(\delta^+(v)), \quad \forall v \in S$$

which are known to strengthen the LP-values of Steiner tree problems, see [9],[12]. We are currently working on a version of these inequalities which incorporates y^h variables, similar to inequalities (NACut).

The branching priorities are set as following: Each variable y_v is assigned priority $p_v + 1 + H$, each variable y_v^h gets priority $H - h$ and arc variables are assigned priority zero. This setting is chosen, since we conjecture that the most important decision in the STPRBH is to decide, which nodes, especially nodes with positive revenue, are in the solution. Moreover, if a node v lies on a layer near the root node, it will greater influence the structure of the solution, than v lying on a layer near H.

Primal Heuristic Our primal heuristic is a modification of the improved version Prim-I [13] of the well-known Prim-based Steiner tree heuristic [14]. The heuristic works similar to Prim's minimum spanning tree algorithm [15], which starts with some node (the root node r, in our case) and then greedily grows the solution tree Sol by adding the node $v \notin Sol$, with minimum connection cost to Sol, i.e., the minimum cost edge $e = argmin\{c_{e=vs} : (v,s) : v \in V \setminus Sol, s \in Sol\}$, until all nodes are added. In the Steiner tree case, the solution Sol is grown by greedily adding terminal nodes $t \notin Sol$, with minimum connection cost, the connection cost is now not the cost of a single edge, but the cost from

Sol to the terminal. When adding the chosen terminal to Sol, all the nodes on the paths are also added to Sol. We modified the algorithm Prim-I for the STPRBH, by taking the hop-limit and the budget-limit into account. This can be easily achieved, since Prim-I works similar to Dijsktra's shortest path algorithm: Whenever an arc is going to be considered as part of a shortest path to a terminal, we check, if the hop-constrained is still fulfilled after adding the arc (note that for this check, the value $H - dist(v, T)$ can be used, instead of the hop-limit), if not, we ignore the connection offered by the arc. The budget-constraint is checked, whenever a terminal is added, if it would be violated, we stop the algorithm. When using this algorithm as primal heuristic, we set the arc weights to $\bar{c}_a = c_a(1 - \tilde{x}_a)$, where $\tilde{x}_a$ is the current LP-value of variable x_a. We have also experimented to take the information offered by $\tilde{y}_v^h$ into account for the arc weights, but in general this produced worse results. The algorithm is also used as starting heuristic, in this case, the original arc weights c_a are used.

Instances We tested our algorithm on the instances provided on the DIMACS-homepage. These instances have been proposed by [2] and [8]. Both are based on the graphs from the sets B and C of the Steiner tree problem (STP) graphs of OR-lib [3]. The transformation into STPRBH-instances is done as follows:

- terminal nodes from the STP are used as profitable nodes by associating a random profit to it
- the budget B is determined as $\sum_{e \in E} c_e/b$, where b is a given divisor
- a hop-limit H is given

Using this transformation, the following set of instances have been created in [2] and [8] (see Table 1).

Table 1: Instances from the DIMACS-homepage (information taken from the file by Zhang-Hua Fu and Jin-Kao Hao). Instances of the upper group have been proposed by [2], the remaining ones by [8].

graphs	p	b	H	number of inst.
B1-B18	[1-100]	5, 20	3, 6, 9, 12	144
C01-C05	[1,10], [1,100]	10, 30	5, 15, 25	60
C06-C10	[1,10], [1,100]	20, 50	5, 15, 25	60
C10-C15	[1,10], [1,100]	10, 100	5, 15, 25	60
C15-C20	[1,10], [1,100]	100, 200	5, 15, 25	60
C16	[1,10], [1,100]	10000	5, 15, 25	6
C17	[1,10], [1,100]	5000	5, 15, 25	6
C18-C20	[1,10], [1,100]	1000	5, 15, 25	18

Results We have set a time limit of 1 000 seconds for our testruns. All instances of set B can be solved in a few seconds with our algorithm using both IMPROVED or BASIC, so we do not show the results here. Detailed results for the C instances can be found in the Appendix: Table 2 provides the results for the new instances C from [8], whereas Tables 3 – 6 give the results for the instances C from [2]. Each table reports the obtained solution value ($sol.\ val$), which is shown in bold, if we have been able to prove optimality. The obtained upper bound (UB) is also given, note that the instances are all integral, thus we used $UB - sol.\ val < 1$ as stopping criterion. In addition, the gap after the timelimit is given [$Gap\%$], as well as the root gap [$RGap\%$]. This gaps are given with respect to the best found solution value. If we have $UB - sol.\ val < 1$, *opt* is written instead. Note that this does not mean that optimality is proven in the root node, since the optimal solution may have not been found yet. On the other hand, CPLEX in some cases is able to use problem specific information to prove optimality even if the root gap is greater than one. The time ($t[s]$) needed to prove optimality is also reported. If we were not able to prove optimality within our time limit of 1 000 seconds, the corresponding entry in the table is "-". The entry $tbest[s]$ contains the time when the best solution has been found and *nodes* gives the number of nodes in the branch-and-cut tree.

Concerning the set of instances from [2], before our study, only 312 out of 384 instances of this set have been solved to optimality (instances based on B and $C1$–$C5$ were solved to optimality by branch-and-cut algorithms from [2], and for the remaining 108 instances, heuristics from [7],[8] found solutions with objective value similar to the sum of all revenues). Using our new approach IMPROVED, we have singinificantly improved these results: We proved the optimality for 378 instances, and for the remaining 6 instances of this set, we additionally improve the best known solutions.

For the new instances from [8], 25 out of 30 were solved to optimality using IMPROVED. None of them has been solved to optimality before.

Using the setting BASIC, 368 and 22 instances from [2] and [8], respectively, are solved within the timelimit, thus the valid inequalities used in IMPROVED clearly have a positive effect on the algorithmic performance. Taking a closer look at the results, we observe that for more than 75% of the instances of set C, the approach IMPROVED proves optimality already in the root node. Furthermore, for more than 85% of the instances of set C, optimality is proven within 50 seconds runtime.

3.2 HCSpT

The initial model additionally contains equalities (15), i.e., the constraint that all nodes must be in the solution, and inequalities (o2AHLink). The branching priorities are set in a similar way as for the STPRBH, naturally we do not give priorities to y_v, since these variables are all fixed to one. We applied the following preprocessing from [1]: If $c_{vw} \geq c_{rw}$, then there exists an optimal solution not using the arc (v, w) and thus such an arc can be removed.

Primal Heuristic We use a modified version of Prim's minimum spanning tree algorithm, where we only add an arc if doing so does not violate the hop-constraint. Given an LP-solution $(\tilde{x}, \tilde{y}, \tilde{y}^h)$, we set the arc weights to $\bar{c}_a = c_a(1 - \tilde{x}_a)$ for $\tilde{x}_a > 0$ and to $\bar{c}_a = M$ (with $M = 10000$ in our computations) otherwise. Similar to the STPRBH the heuristic is also used as starting heuristic, and the original weights are taken in this case. Again, trying to incorporate $\tilde{y}^h$ into the arc costs did not lead to promising results in our initial tests.

Instances We tested our algorithm on the instances used in [1]. The instances consist of complete graphs of 20-160 nodes (in steps of 20) plus one fixed root node. Depending on the cost structure, the instances are classified into three sets, TC, TE and TR. The first two sets have Euclidean costs, in the set TC the root is located in the center, and in TE it is located in a corner. In set TR, the edge costs are randomly generated and the largest graph has 60 nodes. The used hoplimit is $H = 3, 4, 5$.

Results For the runs on the instances from [1] reported in this article, a time-limit of 2000 seconds was used. Tables 7 – 9 give the results for TC, TE and TR respectively. We report the same values in each column as for the STPRBH, except that UB is replaced by the best obtained lower bound (since HCSpT is a minimization problem). Moreover, we also provide the optimal solutions as reported in [1]. The gaps in the table are calculated with respect to these optimal values.

Our algorithm solves all instances from the set TR to optimality, moreover, it also manages to solve the smaller instances from TC and TE to optimality. For the larger instances, the gaps obtained after 2 000 seconds are up to 25%, however, some of these larger instances are also hard for [1], with e.g., $TC - 160$, $H = 5$ taking over 10 000 seconds and $TE - 160, H = 5$ taking over 50 000 seconds for proving optimality. Moreover, we are not using the full potential of our model, since we have not developed the general node-arc cuts on the layered graph yet.

4 Conclusion

The power of layered graphs has been recently demonstrated for many problems, including hop- and diameter-constrained spanning trees [1], hop-constrained connected facility location [10], or to problems that involve more general hop- or diameter- constraints (see, e.g., [16], [17]).

In this paper, we proposed a new extended formulation based on a layered graph for hop- and diameter- constrained spanning/Steiner tree problems. In contrast to previous approaches from literature, which use variables associated with arcs of the layered graph, our new model projects out these arc variables and relies only on node variables in the layered graph. Thus, models with significantly less variables can be derived, and it remains our next goal to study the benefits of this new approach for other standard problems from the literature.

Our main computational study has been conducted on the STPRBH, for which we have been able to significantly improve results from the available literature: we prove the optimality for all except eleven out of 414 instances from two different data sets. For the eleven unsolved cases, we provide new best known solutions. In addition, for the HCSpT, we compare our new modeling approach with the state-of-the-art branch-and-cut from [1]. The results indicate that the LP-bounds of our new model still can be improved. The subject of our future study is the investigation of a family of more general cut-set inequalities on the layered graph, with the available (reduced) set of binary variables.

Acknowledgements

The research was supported by the Austrian Research Fund (FWF, Project P 26755-N19)

References

1. 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** (2011) 123–148
2. Costa, A.M., Cordeau, J.F., Laporte, G.: Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. Networks **53**(2) (2009) 141–159
3. Beasley, J.E.: OR-Library: distributing test problems by electronic mail. Journal of the operational research society (1990) 1069–1072
4. Costa, A.M., Cordeau, J.F., Laporte, G.: Fast heuristics for the Steiner tree problem with revenues, budget and hop constraints. European Journal of Operational Research **190**(1) (2008) 68–78
5. Sinnl, M.: Branch-and-price for the Steiner tree problem with revenues, budget and hop constraints. Master's thesis, Vienna University of Technology (2011)
6. Layeb, S.B., Hajri, I., Haouari, M.: Solving the Steiner tree problem with revenues, budget and hop constraints to optimality. In: Modeling, Simulation and Applied Optimization (ICMSAO), 2013 5th International Conference on, IEEE (2013) 1–4
7. Fu, Z.H., Hao, J.K.: Breakout local search for the Steiner tree problem with revenue, budget and hop constraints. European Journal of Operational Research **232**(1) (2014) 209–220
8. Fu, Z.H., Hao, J.K.: Dynamic Programming Driven Memetic Search for the Steiner Tree Problem with Revenues, Budget and Hop Constraints. Technical Report (2014)
9. Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G.W., Mutzel, P., Fischetti, M.: An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. Mathematical Programming **105**(2-3) (2006) 427–449
10. Ljubić, I., Gollowitzer, S.: Layered graph approaches to the hop constrained connected facility location problem. INFORMS Journal on Computing **25**(2) (2013) 256–270
11. Gouveia, L.: Using Hop-Indexed Models For Constrained Spanning and Steiner Tree Models. In Sans, B., Soriano, P., eds.: Telecommunications Network Planning. Centre for Research on Transportation. Springer US (1999) 21–32
12. Koch, T., Martin, A.: Solving Steiner tree problems in graphs to optimality. Networks **32** (1998) 207–232
13. de Aragão, M.P., Werneck, R.F.: On the implementation of MST-based heuristics for the Steiner problem in graphs. In: Algorithm Engineering and Experiments. Springer (2002) 1–15
14. Takahashi, H., Matsuyama, A.: An approximate solution for the Steiner problem in graphs. Math. Japonica **24**(6) (1980) 573–577
15. Prim, R.C.: Shortest connection networks and some generalizations. Bell system technical journal **36**(6) (1957) 1389–1401
16. Gouveia, L., Leitner, M., Ljubić, I.: The two-level diameter constrained spanning tree problem. Mathematical Programming (2012) 1–30
17. Gouveia, L., Leitner, M., Ljubić, I.: Hop constrained Steiner trees with multiple root nodes. European Journal of Operational Research **236**(1) (2014) 100–112

Appendix 1: Detailed Results for STPRBH

Table 2: Results for the instances from the DIMACS challenge: Instances based on graphs $C16 - C20$ from [8], setting: IMPROVED

inst	budget	hop	sol. val	UB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
C16-10	10000	5	**19**	19.00	**opt**	42.11	3.94	3.93	0
C16-10	10000	15	**19**	19.00	**opt**	**opt**	7.76	7.74	0
C16-10	10000	25	**19**	19.00	**opt**	42.11	5.54	5.52	0
C16-100	10000	5	**203**	203.00	**opt**	34.98	3.96	3.95	0
C16-100	10000	15	**203**	203.00	**opt**	34.98	5.57	5.55	0
C16-100	10000	25	**203**	203.00	**opt**	34.98	5.53	5.51	0
C17-10	5000	5	**47**	47.00	**opt**	6.38	9.72	0.30	4
C17-10	5000	15	**50**	50.00	**opt**	0.61	14.86	12.17	0
C17-10	5000	25	**50**	50.00	**opt**	**opt**	22.59	17.74	0
C17-100	5000	5	**481**	481.00	**opt**	4.26	35.13	0.31	4
C17-100	5000	15	**513**	513.00	**opt**	0.52	15.38	11.37	0
C17-100	5000	25	**513**	513.00	**opt**	**opt**	42.54	7.81	0
C18-10	1000	5	318	322.06	1.28	1.36	-	50.87	48
C18-10	1000	15	**341**	341.00	**opt**	0.29	78.32	56.90	21
C18-10	1000	25	**341**	341.00	**opt**	0.46	225.64	171.50	31
C18-100	1000	5	3320	3357.87	1.14	1.34	-	229.13	28
C18-100	1000	15	**3552**	3552.00	**opt**	0.44	519.4	122.62	96
C18-100	1000	25	**3557**	3557.00	**opt**	0.25	262.35	76.24	23
C19-10	1000	5	**404**	404.00	**opt**	**opt**	62.66	58.78	0
C19-10	1000	15	**428**	428.00	**opt**	**opt**	17.2	16.13	0
C19-10	1000	25	**428**	428.00	**opt**	**opt**	26.59	16.58	0
C19-100	1000	5	**4179**	4179.00	**opt**	0.28	194.16	110.98	62
C19-100	1000	15	**4435**	4435.00	**opt**	0.05	38.75	23.03	3
C19-100	1000	25	**4435**	4435.00	**opt**	0.07	98.19	82.45	7
C20-10	1000	5	**460**	460.00	**opt**	0.63	915.89	85.17	196
C20-10	1000	15	504	505.77	0.35	0.4	-	678.96	160
C20-10	1000	25	**506**	506.00	**opt**	**opt**	62.95	54.42	0
C20-100	1000	5	4768	4804.08	0.76	0.83	-	186.12	54
C20-100	1000	15	5222	5255.46	0.64	0.66	-	749.28	103
C20-100	1000	25	**5256**	5256.00	**opt**	0.01	108.25	106.06	1

Table 3: Results for the instances from the DIMACS challenge: Instances based on graphs $C01 - C05$ from [2], setting: IMROVED

inst	budget	hop	sol. val	UB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
C01-10	10	5	**8**	8.00	**opt**	**opt**	0.02	0.02	0
C01-10	30	5	**8**	8.00	**opt**	**opt**	0.02	0.02	0
C01-10	10	15	**27**	27.00	**opt**	**opt**	0.09	0.03	0
C01-10	30	15	**27**	27.00	**opt**	**opt**	0.09	0.03	0
C01-10	10	25	**27**	27.00	**opt**	**opt**	0.21	0.04	0
C01-10	30	25	**27**	27.00	**opt**	**opt**	0.22	0.04	0
C01-100	10	5	**71**	71.00	**opt**	**opt**	0.02	0.02	0
C01-100	30	5	**71**	71.00	**opt**	**opt**	0.02	0.02	0
C01-100	10	15	**274**	274.00	**opt**	**opt**	0.1	0.03	0
C01-100	30	15	**274**	274.00	**opt**	**opt**	0.09	0.03	0
C01-100	10	25	**274**	274.00	**opt**	**opt**	0.21	0.04	0
C01-100	30	25	**274**	274.00	**opt**	**opt**	0.22	0.04	0
C03-10	10	5	**151**	151.00	**opt**	**opt**	0.03	0.02	0
C03-10	30	5	**95**	95.00	**opt**	1.04	0.05	0.03	0
C03-10	10	15	**289**	289.00	**opt**	0.3	4.55	4.54	24
C03-10	30	15	**129**	129.00	**opt**	**opt**	1.89	1.24	0
C03-10	10	25	**289**	289.00	**opt**	0.36	4.67	1.18	1
C03-10	30	25	**129**	129.00	**opt**	**opt**	5.46	5.45	0
C03-100	10	5	**1519**	1519.00	**opt**	**opt**	0.03	0.02	0
C03-100	30	5	**968**	968.00	**opt**	0.69	0.13	0.06	4
C03-100	10	15	**2971**	2971.00	**opt**	0.55	18.29	6.16	109
C03-100	30	15	**1343**	1343.00	**opt**	0.3	1.31	0.03	0
C03-100	10	25	**2979**	2979.00	**opt**	0.42	26.87	5.01	115
C03-100	30	25	**1343**	1343.00	**opt**	0.3	4.07	0.05	0
C04-10	10	5	**115**	115.00	**opt**	**opt**	0.03	0.02	0
C04-10	30	5	**84**	84.00	**opt**	4.24	0.03	0.02	0
C04-10	10	15	**336**	336.00	**opt**	1.51	157.63	6.22	522
C04-10	30	15	**134**	134.00	**opt**	1.4	4.73	4.52	0
C04-10	10	25	**341**	341.00	**opt**	0.02	3.91	2.56	0
C04-10	30	25	**136**	136.00	**opt**	0.78	3.3	1.97	0
C04-100	10	5	**1148**	1148.00	**opt**	**opt**	0.03	0.02	0
C04-100	30	5	**854**	854.00	**opt**	**opt**	0.03	0.02	0
C04-100	10	15	**3458**	3458.00	**opt**	1.31	392.39	18.39	941
C04-100	30	15	**1380**	1380.00	**opt**	1.48	6.8	4.21	32
C04-100	10	25	**3504**	3504.00	**opt**	**opt**	10	0.66	0
C04-100	30	25	**1396**	1396.00	**opt**	0.34	8.75	2.68	4
C05-10	10	5	**258**	258.00	**opt**	**opt**	0.05	0.03	0
C05-10	30	5	**154**	154.00	**opt**	**opt**	0.05	0.03	0
C05-10	10	15	**494**	494.00	**opt**	0.61	22.24	21.33	49
C05-10	30	15	**182**	182.00	**opt**	1.4	6.77	6.75	19
C05-10	10	25	**495**	495.00	**opt**	0.34	31.68	17.95	52
C05-10	30	25	**183**	183.00	**opt**	0.64	6.19	2.70	0
C05-100	10	5	**2600**	2600.00	**opt**	**opt**	0.04	0.02	0
C05-100	30	5	**1584**	1584.00	**opt**	**opt**	0.04	0.02	0
C05-100	10	15	**5032**	5032.00	**opt**	0.25	77.5	16.47	386
C05-100	30	15	**1857**	1857.00	**opt**	1.02	5.9	3.29	27
C05-100	10	25	**5044**	5044.00	**opt**	0.19	132.36	39.65	283
C05-100	30	25	**1860**	1860.00	**opt**	0.81	87.09	8.94	302

Table 4: Results for the instances from the DIMACS challenge: Instances based on graphs $C06 - C10$ from [2], setting: IMPROVED

inst	budget	hop	sol. val	UB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
C06-10	20	5	**27**	27.00	**opt**	**opt**	0.03	0.02	0
C06-10	50	5	**27**	27.00	**opt**	**opt**	0.04	0.03	0
C06-10	20	15	**27**	27.00	**opt**	**opt**	0.2	0.04	0
C06-10	50	15	**27**	27.00	**opt**	**opt**	0.22	0.05	0
C06-10	20	25	**27**	27.00	**opt**	**opt**	0.46	0.07	0
C06-10	50	25	**27**	27.00	**opt**	**opt**	0.45	0.06	0
C06-100	20	5	**274**	274.00	**opt**	**opt**	0.03	0.02	0
C06-100	50	5	**274**	274.00	**opt**	**opt**	0.03	0.02	0
C06-100	20	15	**274**	274.00	**opt**	**opt**	0.22	0.05	0
C06-100	50	15	**274**	274.00	**opt**	**opt**	0.23	0.06	0
C06-100	20	25	**274**	274.00	**opt**	**opt**	0.46	0.06	0
C06-100	50	25	**274**	274.00	**opt**	**opt**	0.44	0.06	0
C07-10	20	5	**49**	49.00	**opt**	**opt**	0.04	0.02	0
C07-10	50	5	**49**	49.00	**opt**	**opt**	0.04	0.02	0
C07-10	20	15	**59**	59.00	**opt**	**opt**	0.22	0.05	0
C07-10	50	15	**59**	59.00	**opt**	**opt**	0.21	0.04	0
C07-10	20	25	**59**	59.00	**opt**	**opt**	0.46	0.06	0
C07-10	50	25	**59**	59.00	**opt**	**opt**	0.46	0.06	0
C07-100	20	5	**503**	503.00	**opt**	**opt**	0.03	0.02	0
C07-100	50	5	**503**	503.00	**opt**	**opt**	0.03	0.02	0
C07-100	20	15	**604**	604.00	**opt**	**opt**	0.23	0.05	0
C07-100	50	15	**604**	604.00	**opt**	**opt**	0.22	0.05	0
C07-100	20	25	**604**	604.00	**opt**	**opt**	0.44	0.06	0
C07-100	50	25	**604**	604.00	**opt**	**opt**	0.47	0.06	0
C08-10	20	5	**230**	230.00	**opt**	0.22	0.08	0.06	0
C08-10	50	5	**116**	116.00	**opt**	0.98	0.13	0.07	0
C08-10	20	15	**331**	331.00	**opt**	0.38	55.19	54.23	55
C08-10	50	15	**171**	171.00	**opt**	1	13.27	12.59	27
C08-10	20	25	**332**	332.00	**opt**	0.08	4.68	4.04	0
C08-10	50	25	**172**	172.00	**opt**	0.42	6.17	5.46	0
C08-100	20	5	**2380**	2380.00	**opt**	**opt**	0.07	0.04	0
C08-100	50	5	**1216**	1216.00	**opt**	0.96	0.21	0.07	17
C08-100	20	15	3431	3452.06	0.61	0.77	-	41.01	209
C08-100	50	15	**1776**	1776.00	**opt**	1.23	23.82	12.50	62
C08-100	20	25	**3455**	3455.00	**opt**	0.07	27.49	24.09	14
C08-100	50	25	**1792**	1792.00	**opt**	0.34	19.23	14.57	11
C09-10	20	5	**304**	304.00	**opt**	0.24	0.9	0.63	0
C09-10	50	5	**149**	149.00	**opt**	0.82	0.73	0.20	12
C09-10	20	15	381	384.93	1.03	1.05	-	87.23	151
C09-10	50	15	**185**	185.00	**opt**	0.92	32.58	19.37	44
C09-10	20	25	**385**	385.00	**opt**	**opt**	28.15	28.14	6
C09-10	50	25	**187**	187.00	**opt**	0.44	8.11	4.66	0
C09-100	20	5	**3133**	3133.00	**opt**	0.1	0.81	0.34	7
C09-100	50	5	**1563**	1563.00	**opt**	0.26	0.5	0.22	0
C09-100	20	15	3945	3964.69	0.5	0.76	-	30.84	338
C09-100	50	15	**1906**	1906.00	**opt**	1.56	425.42	18.05	616
C09-100	20	25	**3974**	3974.00	**opt**	0.06	7.54	6.07	0
C09-100	50	25	**1933**	1933.00	**opt**	0.15	21.3	4.49	6
C10-10	20	5	**391**	391.00	**opt**	**opt**	0.28	0.22	0
C10-10	50	5	**185**	185.00	**opt**	1.22	0.25	0.15	0
C10-10	20	15	565	580.58	2.76	2.95	-	131.22	97
C10-10	50	15	**257**	257.00	**opt**	0.39	6.95	1.29	10
C10-10	20	25	**580**	580.00	**opt**	0.29	345.85	343.30	70
C10-10	50	25	**258**	258.00	**opt**	0.29	8.57	7.12	0
C10-100	20	5	**4096**	4096.00	**opt**	**opt**	0.17	0.09	0
C10-100	50	5	**1940**	1940.00	**opt**	0.13	0.75	0.15	5
C10-100	20	15	5849	5990.03	2.41	2.47	-	236.78	108
C10-100	50	15	**2657**	2657.00	**opt**	0.52	20.68	2.96	18
C10-100	20	25	5972	5991.00	0.32	0.34	-	737.49	152
C10-100	50	25	**2683**	2683.00	**opt**	0.1	6.83	5.40	0

Table 5: Results for the instances from the DIMACS challenge: Instances based on graphs $C11 - C15$ from [2], setting: IMPROVED

inst	budget	hop	sol. val	UB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
C11-10	20	5	**27**	27.00	**opt**	**opt**	0.13	0.06	0
C11-10	100	5	**27**	27.00	**opt**	**opt**	0.13	0.06	0
C11-10	20	15	**27**	27.00	**opt**	**opt**	0.42	0.08	0
C11-10	100	15	**27**	27.00	**opt**	**opt**	0.44	0.09	0
C11-10	20	25	**27**	27.00	**opt**	**opt**	0.84	0.10	0
C11-10	100	25	**27**	27.00	**opt**	**opt**	0.78	0.11	0
C11-100	20	5	**274**	274.00	**opt**	**opt**	0.13	0.06	0
C11-100	100	5	**274**	274.00	**opt**	**opt**	0.13	0.06	0
C11-100	20	15	**274**	274.00	**opt**	**opt**	0.43	0.09	0
C11-100	100	15	**274**	274.00	**opt**	**opt**	0.44	0.09	0
C11-100	20	25	**274**	274.00	**opt**	**opt**	0.85	0.12	0
C11-100	100	25	**274**	274.00	**opt**	**opt**	0.79	0.12	0
C12-10	20	5	**59**	59.00	**opt**	**opt**	0.16	0.06	0
C12-10	100	5	**59**	59.00	**opt**	**opt**	0.17	0.06	0
C12-10	20	15	**59**	59.00	**opt**	**opt**	0.44	0.08	0
C12-10	100	15	**59**	59.00	**opt**	**opt**	0.42	0.08	0
C12-10	20	25	**59**	59.00	**opt**	**opt**	0.82	0.11	0
C12-10	100	25	**59**	59.00	**opt**	**opt**	0.83	0.11	0
C12-100	20	5	**604**	604.00	**opt**	**opt**	0.16	0.06	0
C12-100	100	5	**604**	604.00	**opt**	**opt**	0.16	0.06	0
C12-100	20	15	**604**	604.00	**opt**	**opt**	0.45	0.09	0
C12-100	100	15	**604**	604.00	**opt**	**opt**	0.42	0.08	0
C12-100	20	25	**604**	604.00	**opt**	**opt**	0.8	0.11	0
C12-100	100	25	**604**	604.00	**opt**	**opt**	0.83	0.11	0
C13-10	20	5	**439**	439.00	**opt**	**opt**	0.23	0.07	0
C13-10	100	5	**257**	257.00	**opt**	0.23	3.42	3.21	0
C13-10	20	15	**439**	439.00	**opt**	**opt**	0.49	0.09	0
C13-10	100	15	**319**	319.00	**opt**	0.27	20.99	19.00	1
C13-10	20	25	**439**	439.00	**opt**	**opt**	0.9	0.12	0
C13-10	100	25	**319**	319.00	**opt**	0.4	23.25	10.07	0
C13-100	20	5	**4463**	4463.00	**opt**	**opt**	0.22	0.06	0
C13-100	100	5	**2653**	2653.00	**opt**	0.11	8.07	4.91	0
C13-100	20	15	**4463**	4463.00	**opt**	**opt**	0.47	0.09	0
C13-100	100	15	**3312**	3312.00	**opt**	0.14	31.11	9.62	10
C13-100	20	25	**4463**	4463.00	**opt**	**opt**	0.88	0.12	0
C13-100	100	25	**3317**	3317.00	**opt**	0.09	24.73	24.71	0
C14-10	20	5	**648**	648.00	**opt**	**opt**	0.8	0.79	0
C14-10	100	5	**373**	373.00	**opt**	0.11	7.55	2.08	0
C14-10	20	15	**648**	648.00	**opt**	**opt**	0.52	0.09	0
C14-10	100	15	**404**	404.00	**opt**	**opt**	8.57	7.18	0
C14-10	20	25	**648**	648.00	**opt**	**opt**	0.95	0.12	0
C14-10	100	25	**404**	404.00	**opt**	**opt**	5.51	4.54	0
C14-100	20	5	**6566**	6566.00	**opt**	**opt**	0.83	0.82	0
C14-100	100	5	**3887**	3887.00	**opt**	0.01	7.17	5.21	0
C14-100	20	15	**6566**	6566.00	**opt**	**opt**	0.51	0.09	0
C14-100	100	15	**4205**	4205.00	**opt**	**opt**	3.53	1.75	0
C14-100	20	25	**6566**	6566.00	**opt**	**opt**	0.98	0.13	0
C14-100	100	25	**4205**	4205.00	**opt**	**opt**	7.22	7.20	0
C15-10	20	5	**1248**	1248.00	**opt**	**opt**	36.06	30.52	7
C15-10	100	5	**480**	480.00	**opt**	0.15	3.6	3.13	0
C15-10	20	15	**1248**	1248.00	**opt**	**opt**	0.54	0.10	0
C15-10	100	15	**568**	568.00	**opt**	0.29	74.14	70.20	61
C15-10	20	25	**1248**	1248.00	**opt**	**opt**	1.03	0.13	0
C15-10	100	25	**569**	569.00	**opt**	0.18	22.79	5.73	0
C15-100	20	5	**12533**	12533.00	**opt**	**opt**	48.41	40.43	6
C15-100	100	5	**5000**	5000.00	**opt**	**opt**	2.84	1.74	0
C15-100	20	15	**12533**	12533.00	**opt**	**opt**	0.57	0.11	0
C15-100	100	15	**5889**	5889.00	**opt**	0.32	421.84	161.32	434
C15-100	20	25	**12533**	12533.00	**opt**	**opt**	0.99	0.13	0
C15-100	100	25	**5905**	5905.00	**opt**	**opt**	23.52	22.49	0

Table 6: Results for the instances from the DIMACS challenge: Instances based on graphs $C16 - C20$ from [2], setting: IMPROVED

inst	budget	hop	sol. val	UB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
C16-10	100	5	**27**	27.00	**opt**	**opt**	1	0.30	0
C16-10	200	5	**27**	27.00	**opt**	**opt**	0.98	0.28	0
C16-10	100	15	**27**	27.00	**opt**	**opt**	1.91	0.33	0
C16-10	200	15	**27**	27.00	**opt**	**opt**	1.85	0.33	0
C16-10	100	25	**27**	27.00	**opt**	**opt**	3.05	0.40	0
C16-10	200	25	**27**	27.00	**opt**	**opt**	3.04	0.41	0
C16-100	100	5	**274**	274.00	**opt**	**opt**	0.95	0.28	0
C16-100	200	5	**274**	274.00	**opt**	**opt**	0.96	0.27	0
C16-100	100	15	**274**	274.00	**opt**	**opt**	1.87	0.33	0
C16-100	200	15	**274**	274.00	**opt**	**opt**	1.94	0.36	0
C16-100	100	25	**274**	274.00	**opt**	**opt**	2.8	0.39	0
C16-100	200	25	**274**	274.00	**opt**	**opt**	2.82	0.37	0
C17-10	100	5	**59**	59.00	**opt**	**opt**	0.95	0.26	0
C17-10	200	5	**59**	59.00	**opt**	**opt**	0.99	0.29	0
C17-10	100	15	**59**	59.00	**opt**	**opt**	1.93	0.34	0
C17-10	200	15	**59**	59.00	**opt**	**opt**	1.95	0.34	0
C17-10	100	25	**59**	59.00	**opt**	**opt**	2.99	0.41	0
C17-10	200	25	**59**	59.00	**opt**	**opt**	2.92	0.41	0
C17-100	100	5	**604**	604.00	**opt**	**opt**	1.02	0.29	0
C17-100	200	5	**604**	604.00	**opt**	**opt**	1.08	0.31	0
C17-100	100	15	**604**	604.00	**opt**	**opt**	1.93	0.35	0
C17-100	200	15	**604**	604.00	**opt**	**opt**	1.94	0.34	0
C17-100	100	25	**604**	604.00	**opt**	**opt**	2.79	0.38	0
C17-100	200	25	**604**	604.00	**opt**	**opt**	2.95	0.40	0
C18-10	100	5	**439**	439.00	**opt**	**opt**	1.09	0.29	0
C18-10	200	5	**439**	439.00	**opt**	**opt**	1.18	0.31	0
C18-10	100	15	**439**	439.00	**opt**	**opt**	2.07	0.35	0
C18-10	200	15	**439**	439.00	**opt**	**opt**	2.08	0.35	0
C18-10	100	25	**439**	439.00	**opt**	**opt**	3.09	0.42	0
C18-10	200	25	**439**	439.00	**opt**	**opt**	3.25	0.42	0
C18-100	100	5	**4463**	4463.00	**opt**	**opt**	1.17	0.30	0
C18-100	200	5	**4463**	4463.00	**opt**	**opt**	1.17	0.31	0
C18-100	100	15	**4463**	4463.00	**opt**	**opt**	2	0.33	0
C18-100	200	15	**4463**	4463.00	**opt**	**opt**	2.03	0.34	0
C18-100	100	25	**4463**	4463.00	**opt**	**opt**	3.22	0.44	0
C18-100	200	25	**4463**	4463.00	**opt**	**opt**	3.38	0.47	0
C19-10	100	5	**648**	648.00	**opt**	**opt**	1.22	0.32	0
C19-10	200	5	**648**	648.00	**opt**	**opt**	1.21	0.30	0
C19-10	100	15	**648**	648.00	**opt**	**opt**	2.21	0.37	0
C19-10	200	15	**648**	648.00	**opt**	**opt**	2.28	0.38	0
C19-10	100	25	**648**	648.00	**opt**	**opt**	3.5	0.45	0
C19-10	200	25	**648**	648.00	**opt**	**opt**	3.45	0.45	0
C19-100	100	5	**6566**	6566.00	**opt**	**opt**	1.26	0.33	0
C19-100	200	5	**6566**	6566.00	**opt**	**opt**	1.3	0.33	0
C19-100	100	15	**6566**	6566.00	**opt**	**opt**	2.32	0.39	0
C19-100	200	15	**6566**	6566.00	**opt**	**opt**	2.18	0.38	0
C19-100	100	25	**6566**	6566.00	**opt**	**opt**	3.33	0.43	0
C19-100	200	25	**6566**	6566.00	**opt**	**opt**	3.24	0.41	0
C20-10	100	5	**1248**	1248.00	**opt**	**opt**	1.44	0.35	0
C20-10	200	5	**1248**	1248.00	**opt**	**opt**	1.54	0.38	0
C20-10	100	15	**1248**	1248.00	**opt**	**opt**	2.61	0.43	0
C20-10	200	15	**1248**	1248.00	**opt**	**opt**	2.62	0.45	0
C20-10	100	25	**1248**	1248.00	**opt**	**opt**	3.68	0.47	0
C20-10	200	25	**1248**	1248.00	**opt**	**opt**	3.73	0.48	0
C20-100	100	5	**12533**	12533.00	**opt**	**opt**	1.47	0.35	0
C20-100	200	5	**12533**	12533.00	**opt**	**opt**	1.49	0.36	0
C20-100	100	15	**12533**	12533.00	**opt**	**opt**	2.61	0.44	0
C20-100	200	15	**12533**	12533.00	**opt**	**opt**	2.41	0.40	0
C20-100	100	25	**12533**	12533.00	**opt**	**opt**	3.68	0.49	0
C20-100	200	25	**12533**	12533.00	**opt**	**opt**	3.79	0.49	0

Appendix 2: Detailed Results for HCSpT

Table 7: Results for the TC-instances from [1], setting: IMPROVED (except for TC80, H=4; TC80, H=5; TC100, H=3, where BASIC is reported, due to exceeded memory limit by IMPROVED).

inst	hop	best	sol. val	LB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
20	3	340	**340**	340.00	**opt**	0.47	0.08	0.08	0
20	4	318	**318**	318.00	**opt**	0.33	0.08	0.04	0
20	5	312	**312**	312.00	**opt**	**opt**	0.06	0.02	0
40	3	609	**609**	609.00	**opt**	0.12	1.23	0.73	0
40	4	548	**548**	548.00	**opt**	1.21	3.24	0.69	43
40	5	522	**522**	522.00	**opt**	1.06	2.53	0.51	30
60	3	866	**866**	866.00	**opt**	1.01	8.87	0.88	55
60	4	781	**781**	781.00	**opt**	3.05	100.21	52.59	941
60	5	734	**734**	734.00	**opt**	2.63	830.15	86.15	5178
80	3	1072	**1072**	1072.00	**opt**	3.35	694.51	0.08	5726
80	4	981	1067	857.11	14.46	14.85	-	749.61	7469
80	5	922	1059	830.40	11.03	11.08	-	692.78	6583
100	3	1259	1306	1161.04	8.44	8.92	-	1208.30	4813
100	4	1166	1206	1071.94	8.77	9.91	-	813.41	1469
100	5	1104	1168	1032.10	6.97	7.97	-	30.70	828
120	3	1059	1069	1022.66	3.55	4.64	-	372.46	1707
120	4	926	988	845.75	9.49	9.72	-	1467.73	384
120	5	853	1082	779.39	9.44	9.58	-	56.76	147
160	3	1357	1462	1233.69	10	10.02	-	730.52	177
160	4	1133	1424	965.21	17.38	17.45	-	1244.81	100
160	5	1039	1328	877.55	18.4	18.51	-	234.37	70

Table 8: Results for the TE-instances from [1]

inst	hop	best	sol. val	LB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
20	3	449	**449**	449.00	**opt**	3.07	0.82	0.79	20
20	4	385	**385**	385.00	**opt**	3.13	1.44	0.96	52
20	5	366	**366**	366.00	**opt**	5.43	3.78	2.90	146
40	3	708	**708**	708.00	**opt**	5.49	18.45	0.64	331
40	4	627	**627**	627.00	**opt**	9.22	463.52	290.26	1980
40	5	590	596	558.42	5.66	9.97	-	279.00	2453
60	3	1525	**1525**	1525.00	**opt**	6.41	245.73	0.05	1583
60	4	1336	1377	1204.34	10.93	12.24	-	1384.93	1000
60	5	1225	1444	1089.32	12.46	12.58	-	38.75	163
80	3	1806	1812	1765.75	2.28	8.27	-	0.10	5612
80	4	1558	1845	1360.64	14.5	14.90	-	650.97	115
80	5	1442	1760	1255.68	14.84	15.22	-	19.58	82
100	3	2092	2104	1944.86	7.57	10.11	-	510.39	1625
100	4	1788	2225	1533.34	16.61	16.84	-	178.65	112
100	5	1625	1931	1392.52	16.69	17.06	-	810.46	102
120	3	1267	1376	1142.98	10.85	10.94	-	1611.41	296
120	4	1074	1321	884.01	21.49	21.63	-	1797.18	132
120	5	969	1245	800.58	21.04	21.23	-	498.93	108
160	3	1496	1618	1316.17	13.66	13.76	-	919.67	101
160	4	1229	1678	993.42	23.71	23.80	-	1824.09	58
160	5	1107	1537	888.47	24.6	24.80	-	549.02	46

Table 9: Results for the TR-instances from [1]

inst	hop	best	sol. val	LB	Gap [%]	RGap [%]	t [s]	tbest [s]	nodes
20	3	168	**168**	168	**opt**	**opt**	0.03	0.01	0
20	4	146	**146**	146	**opt**	4.53	0.03	0.01	0
20	5	137	**137**	137	**opt**	**opt**	0.02	0.01	0
40	3	176	**176**	176	**opt**	2.21	0.36	0.35	0
40	4	149	**149**	149	**opt**	0.64	0.93	0.46	0
40	5	139	**139**	139	**opt**	0.66	1.42	0.69	4
60	3	213	**213**	213	**opt**	0.77	1.83	1.72	0
60	4	152	**152**	152	**opt**	1.17	8.11	0.95	9
60	5	124	**124**	124	**opt**	0.05	6.04	1.67	0