On Solving the Rooted Delay- and Delay-Variation-Constrained Steiner Tree Problem

Mario Ruthmair and Günther R. Raidl

Vienna University of Technology, Vienna, Austria
Institute of Computer Graphics and Algorithms
{ruthmair,raidl}@ads.tuwien.ac.at

Abstract. We present mixed integer programming approaches for optimally solving a combinatorial optimization problem arising in network design with additional quality of service constraints. The rooted delay- and delay-variation-constrained Steiner tree problem asks for a cost-minimal Steiner tree satisfying delay-constraints from source to terminals and a maximal variation-bound between particular terminal path-delays. Our MIP models are based on multi-commodity-flows and a layered graph transformation. For the latter model we propose some new sets of valid inequalities and an efficient separation method. Presented experimental results indicate that our layered graph approaches clearly outperform the flow-based model.

1 Introduction

We consider problems arising in client-server network design with additional quality of service (QoS) constraints. In VoIP and video conferencing multicast scenarios it is not only important that all participants receive the information from the central server within a given time limit but also nearly at the same time. Otherwise upcoming race conditions possibly result in misunderstandings between the clients. In database replication scenarios it is necessary to guarantee the consistency of all mirroring databases. Thus, if updates have to be deployed the time interval between the first and the last client database applying the changes should be within a predefined limit. Buffering information at the server or intermediate nodes in the network shall be avoided as in general it would increase the total delay and requires the repeated sending of the same data, annihilating the advantage of distributing information over a multicast tree. Finally, buffering at the clients is not always a choice since in some online applications, e.g. gaming and stocktrading, competing users may benefit from receiving information earlier than others and thus may circumvent the local data retention. Beside these QoS constraints minimizing the total cost of used connections is in most cases a desired criterium. These problems can be modeled as *rooted delay- and delay-variation-constrained Steiner tree (RDDVCST) problem.*

More formally, we are given an undirected graph $G = (V, E)$ with node set V, a fixed root node $s \in V$, set $R \subseteq V \setminus \{s\}$ of terminal or required nodes,

A.R. Mahjoub et al. (Eds.): ISCO 2012, LNCS 7422, pp. 225–236, 2012.

set $S = V \setminus (R \cup \{s\})$ of optional Steiner nodes, edge set E, a cost function $c : E \rightarrow \mathbb{Z}^+$, a delay function $d : E \rightarrow \mathbb{Z}^+$, a delay-bound $B \in \mathbb{Z}^+$ and a delay-variation-bound $D \in \mathbb{Z}_0^+$. An optimal solution to the RDDVCST problem is a Steiner tree $T = (V^T, E^T),\ s \in V^T,\ R \subset V^T \subseteq V,\ E^T \subseteq E$, with minimum cost $c(T) = \sum_{e \in E^T} c_e$, satisfying the delay-constraints

$$d_v^T = \sum_{e \in P_T(s,v)} d_e \leq B,\ \forall v \in R, \tag{1}$$

where $P_T(s,v)$ denotes the unique path from root s to node v in tree T, and d_v^T the total delay of this path. We further limit the difference between the path-delays to any two terminal nodes by the constraint

$$\max_{u,v \in R} |d_u^T - d_v^T| \leq D. \tag{2}$$

Here, we present two exact mixed integer programming (MIP) approaches: a multi-commodity-flow (MCF) model and a delay-indexed formulation on a corresponding layered graph. The latter model is tightened by valid inequalities based on well-known directed connection cuts and a new set of constraints utilizing the delay-variation-bound. We further show that the MCF model is not competitive regarding the practical computation times.

2 Previous and Related Work

Rouskas and Baldine [12] introduce a variant of the RDDVCST problem called *delay- and delay-variation-bounded multicast tree (DVBMT)* problem. In it the aim is to just find a feasible tree satisfying both the delay- and delay-variation-constraints without considering edge costs at all. As even this decision version is NP-hard, this also holds for the RDDVCST problem. To solve the DVBMT problem the authors present a construction heuristic with relatively high runtime complexity starting with a feasible path to one terminal node and iteratively connecting the rest of the terminals in feasible ways as long as possible by computing k-shortest-delay-paths. Haberman and Rouskas [6] tackle the RDDVCST problem for the first time and present a heuristic similar to the one in [12] but additionally considering edge costs. Lee et al. [9] provide another construction heuristic: first, the shortest-delay-paths to all terminals are combined to form a tree naturally satisfying the delay-constraint. Second, tree costs are reduced possibly violating delay- and delay-variation-constraints. Not feasibly connected terminals are then removed and re-added to the tree by low-delay paths. Low et al. [11] present a two phase construction approach: in the first phase a tree is obtained by only considering the costs and the delay-constraint. If the delay-variation-constraint is violated in this solution the second phase searches for alternative paths in a distributed way. Sheu et al. [15] improve the worst-case time complexity of the heuristic in [12] for the DVBMT problem still obtaining high quality solutions in the sense that the delay-variation is quite low.

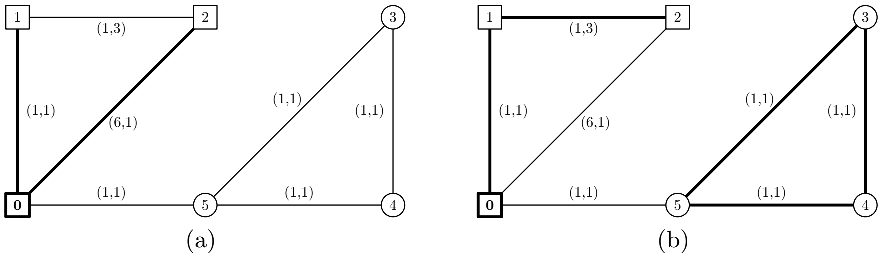

Fig. 1. (a) Example graph G with edge labels (c_e, d_e) and root node 0. Squared nodes denote terminal nodes and bold edges show the optimal solution for $B = 4,\ D = 0$, with $c(T) = 7$. (b) The optimal solution to model *MCF* has costs $c(T) = 5$ but is infeasible for the RDDVCST problem.

Zhang et al. [8] propose a simulated annealing approach for the RDDVCST problem using a path-based solution encoding scheme and a path-exchange neighborhood only allowing feasible moves.

To the best of our knowledge only one MIP formulation exists so far for another problem variant in which the delay-variation is minimized: Sheu et al. [16] present an MCF formulation, which we revise and adapt to the RDDVCST problem in Section 3. Omitting the delay-variation-constraint yields the more prominent *rooted delay-constrained Steiner tree (RDCST) problem* which has been tackled among others by us in [14]. We proposed a transformation to a layered graph allowing a strong formulation. However, we argue that the layered graph can become very large due to its dependency on the delay-bound B, further possibly resulting in a computationally intractable MIP model. Therefore, we suggested a so-called *adaptive layers framework (ALF)* in [14] which dynamically approximates the structure of the full layered graph and iteratively computes lower and upper bounds to an optimal solution. Unfortunately, ALF cannot be applied in a straight-forward way to the RDDVCST problem.

3 Multi-commodity-Flow Formulation

We define a directed graph $G' = (V, A)$ originating from graph G with arc set $A = \{(s, v) \mid \{s, v\} \in E\} \cup \{(u, v), (v, u) \mid \{u, v\} \in E,\ u, v \neq s\}$. Arc delay and cost values are adopted from the corresponding edges. Following Gouveia [3], an MCF model for our problem on a directed graph provides the same strength and needs in general less constraints than on the corresponding undirected graph. Because of this and since a solution to the RDDVCST problem can be modeled as an equivalent Steiner arborescence directed out of root node s, we use G' as base graph in the rest of the article. Since preprocessing graph G to reduce the problem size is important to decrease runtimes, we eliminate infeasible edges as described in [13,14] for the RDCST problem. However, it is not feasible here to remove suboptimal edges as shown in [13]: In some cases we may have to choose expensive edges with high delays to satisfy the delay-variation-constraint.

Nevertheless, we are able to utilize the delay-variation-bound to further reduce graph G by removing all edges connecting two terminal nodes with $d_e > D$ since they clearly cannot appear in any feasible solution. Additionally, in graph G' we can safely remove all arcs $(u,v) \in A$ with $u \in R$ and $d_{uv} > D$.

We use binary decision variables $x_{uv},\ \forall (u,v) \in A$. Furthermore, real-valued flow variables $f_{uv}^w,\ \forall (u,v) \in A,\ \forall w \in R$, denote the flow on arc (u,v) from root s to terminal w. The minimal path-delay is described by variable $\delta_{\min}$. Model *MCF* is defined as follows:

$$\min \sum_{(u,v)\in A} c_{uv} x_{uv} \tag{3}$$

$$\text{s.t.} \quad \sum_{(u,v)\in A} f_{uv}^w - \sum_{(v,u)\in A} f_{vu}^w = \begin{cases} -1 & \text{if } v = s \\ 1 & \text{if } v = w \\ 0 & \text{else} \end{cases} \qquad \forall w \in R \tag{4}$$

$$\delta_{\min} \le \sum_{(u,v)\in A} d_{uv} f_{uv}^w \le \delta_{\min} + D \qquad \forall w \in R \tag{5}$$

$$\delta_{\min} \in [1, B - D] \tag{6}$$

$$0 \le f_{uv}^w \le x_{uv} \qquad \forall (u,v) \in A,\ \forall w \in R \tag{7}$$

$$x_{uv} \in \{0,1\} \qquad \forall (u,v) \in A \tag{8}$$

Classical flow constraints (4) describe the flow of one commodity for each terminal $w \in R$ originating in root s, possibly passing any nodes in $V \setminus \{s,w\}$, and ending in target node w, respectively. Constraints (5) add up the delays on the path to a terminal and define lower and upper delay-bounds over all required nodes respecting the delay-variation D. Since variable $\delta_{\min}$ is restricted to $[1, B-D]$ the delay-bound B is satisfied implicitly. Finally, linking constraints (7) connect flow and arc variables.

Providing edge costs are strictly positive, objective (3) together with constraints (4), (7) and (8) describe optimal Steiner trees, cf. [3]. However, by adding constraints (5) and (6) detached cycles consisting of Steiner nodes may occur in an optimal solution to model *MCF*, see Fig. 1: arcs $(0,1)$ and $(1,2)$ connect both terminal nodes to the root within the given delay-bound $B = 4$ but result in a delay-variation of $D = 3$. Instead of using optimal arcs $(0,1)$ and $(0,2)$ it is cheaper and feasible in model *MCF* to add a circular flow for terminal 1 on the detached cycle $(3,4,5)$, so $f_{01}^1 = f_{34}^1 = f_{45}^1 = f_{53}^1 = 1$ and $f_{01}^2 = f_{12}^2 = 1$. Due to constraints (5) the "path-delay" to node 1 is now increased to 4 and thus $D = 0$. To prevent infeasible solutions we guarantee root connectivity for all used Steiner nodes. Therefore, we add sets of flow variables and constraints for each Steiner node. But only if there is an incoming arc to a Steiner node the corresponding flow is activated. This finally feasible model *MCF′* extends *MCF* by:

$$\sum_{(u,v)\in A} f_{uv}^w - \sum_{(v,u)\in A} f_{vu}^w = \begin{cases} -\sum_{(u,w)\in A} x_{uw} & \text{if } v = s \\ \sum_{(u,w)\in A} x_{uw} & \text{if } v = w \\ 0 & \text{else} \end{cases} \qquad \forall w \in S \tag{9}$$

$$0 \le f_{uv}^w \le x_{uv} \qquad \forall (u,v) \in A,\ \forall w \in S \tag{10}$$

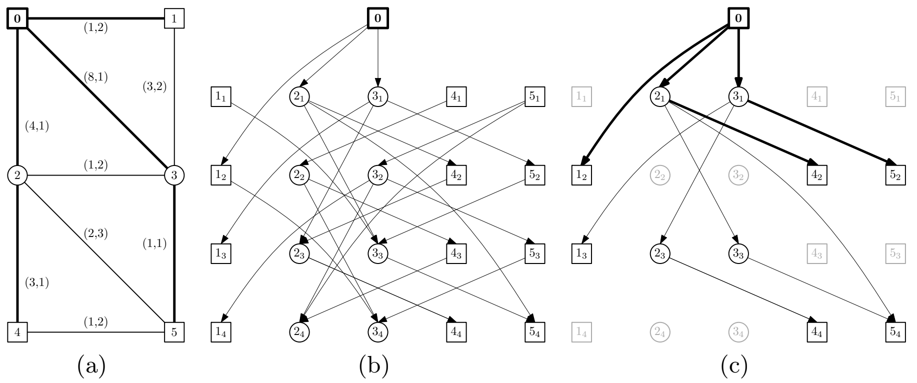

Fig. 2. (a) Example graph G with edge labels (c_e, d_e) and root node 0. Squared nodes denote terminal nodes and bold arcs show the optimal solution for $B = 4,\ D = 1$. Corresponding layered digraph G_L before (b) and after (c) preprocessing (arc costs are omitted).

Flow constraints (9) for Steiner nodes are similar to the counterparts (4) for terminal nodes but extended by in-degree terms to optionally enable or disable the corresponding flows.

4 Layered Graph Transformation

Similarly to [5,14] we transform graph $G' = (V, A)$ to a layered digraph $G_\mathrm{L} = (V_\mathrm{L}, A_\mathrm{L})$ with node set $V_\mathrm{L} = \{s\} \cup \{v_l \mid v \in V \setminus \{s\},\ 1 \leq l \leq B\}$. Thus, we introduce copies of all nodes except the root for each possible delay value. Arc set $A_\mathrm{L} = A^s_\mathrm{L} \cup A^g_\mathrm{L}$ consists of root arcs $A^s_\mathrm{L} = \{(s, v_{d_{sv}}) \mid (s, v) \in A\}$ and general arcs $A^g_\mathrm{L} = \{(u_l, v_{l+d_{uv}}) \mid (u, v) \in A,\ u, v \neq s,\ 1 \leq l \leq B - d_{uv}\}$. Arc delays d_{uv} are not needed in G_L since they are implicitly contained in the layered structure: node v_l in G_L represents node v in G' with $d^T_v = l$ in a solution T. Arc costs in A^s_L and A^g_L are the same as the costs of corresponding arcs in A.

We want to find an arborescence $T_\mathrm{L} = (V^T_\mathrm{L}, A^T_\mathrm{L})$ in G_L with $V^T_\mathrm{L} \subseteq V_\mathrm{L},\ A^T_\mathrm{L} \subseteq A_\mathrm{L}$, rooted in $s \in V^T_\mathrm{L}$, including exactly one node $v_l \in V^T_\mathrm{L}$ for each terminal node $v \in R$ and at most one node $u_l \in V^T_\mathrm{L}$ for each Steiner node $u \in S$, having minimal costs $c(T_\mathrm{L}) = \sum_{(u_k,v_l)\in A^T_\mathrm{L}} c_{uv}$ and satisfying the transformed delay-variation-constraint

$$\max_{u_k, v_l \in V^T_\mathrm{L},\ u,v \in R} |k - l| \leq D. \qquad (11)$$

An optimal arborescence T^*_L in G_L as defined above corresponds to an optimal Steiner arborescence T^* for the RDDVCST problem on G', moreover $c(T^*_\mathrm{L}) = c(T^*)$. A solution T^* in G is obtained from an arborescence T^*_L by simply mapping all nodes $v_l \in V^T_\mathrm{L} \setminus \{s\}$ to v and arcs to edges correspondingly.

Due to its possibly huge size preprocessing in G_L is even more important than in G. The following reduction steps are repeated as long as G_L is modified by one of them:

1. A node $v_l \in V_\mathrm{L},\ v \in R$, is removed if $\exists u \in R \setminus \{v\}$ with $u_k \notin V_\mathrm{L},\ \forall k \in \{l-D, l+D\}$, since v_l cannot appear in any feasible solution.
2. Let $\deg^-(u_k)$ and $\deg^+(u_k)$ denote the in- and outdegree of node u_k, respectively. To partly prevent cycles of length two in G' an arc $(u_k, v_l) \in A_\mathrm{L}$ is removed if $\deg^-(u_k) = 1 \wedge (v_m, u_k) \in A_\mathrm{L}$ or $v \in S \wedge \deg^+(v_l) = 1 \wedge (v_l, u_m) \in A_\mathrm{L}$.
3. If node $v_l \in V_\mathrm{L} \setminus \{s\}$ has no incoming arcs it cannot be reached from s and therefore is removed.
4. If node $v_l \in V_\mathrm{L} \setminus \{s\},\ v \in S$, has no outgoing arcs it is removed since a Steiner node cannot be a leaf in an optimal solution.

These preprocessing rules are able to reduce the number of nodes and arcs significantly especially for instances with a broad range of delay values. Instances with a variation-bound too tight to allow a feasible solution are usually hard to identify. However, our preprocessing procedure is in many of those cases able to detect infeasibility by removing the whole set V_L. Further reduction methods for Steiner trees can be found in [7,10]. See Fig. 2 for an example of layered graph transformation, preprocessing, and solution correspondance.

5 Layered Graph Approaches

The model presented in [14] solves the problem variant without the delay-variation-constraint on layered graph G_L. Here, we revise and extend it by adding an additional set of variables and considering the bounded delay-variation. New continuous variables $y_v^l,\ \forall v_l \in V_\mathrm{L} \setminus \{s\}$, and $x_{uv}^k,\ \forall (u_k, v_l) \in A_\mathrm{L}$, represent nodes and arcs in layered graph G_L, respectively. Model LAY is defined as follows:

$$\min \sum_{(u,v)\in A} c_{uv} x_{uv} \tag{12}$$

$$\text{s.t.} \quad \sum_{v_l \in V_\mathrm{L}} y_v^l = 1 \qquad \forall v \in R \tag{13}$$

$$\sum_{v_l \in V_\mathrm{L}} y_v^l \le 1 \qquad \forall v \in S \tag{14}$$

$$\sum_{(u_k, v_l) \in A_\mathrm{L}} x_{uv}^k = y_v^l \qquad \forall v_l \in V_\mathrm{L} \setminus \{s\} \tag{15}$$

$$\sum_{(u_j, v_k) \in A_\mathrm{L}, u \neq w} x_{uv}^j \ge x_{vw}^k \qquad \forall (v_k, w_l) \in A_\mathrm{L}^g \tag{16}$$

$$x_{sv}^0 = x_{sv} \qquad \forall (s,v) \in A \tag{17}$$

$$\sum_{(u_k, v_l) \in A_\mathrm{L}} x_{uv}^k = x_{uv} \qquad \forall (u,v) \in A,\ u \neq s \tag{18}$$

$$\delta_{\min} \le \sum_{l=1}^{B} l \cdot y_v^l \le \delta_{\min} + D \qquad \forall v \in R \tag{19}$$

$$\delta_{\min} \in [1, B - D] \tag{20}$$
$$x^k_{uv} \geq 0 \qquad \forall (u_k, v_l) \in A_{\mathrm{L}} \tag{21}$$
$$y^l_v \geq 0 \qquad \forall v_l \in V_{\mathrm{L}} \setminus \{s\} \tag{22}$$
$$x_{uv} \in \{0, 1\} \qquad \forall (u, v) \in A \tag{23}$$

Constraints (13) and (14) state that from the set of layered graph nodes corresponding to one particular original node exactly one has to be chosen for required nodes and at most one for Steiner nodes, respectively. Indegree constraints (15) in G_{L} restrict the number of incoming arcs to a layered graph node v_l in dependency of y^l_v to at most one. Since G_{L} is acyclic constraints (16) are enough to ensure connectivity. Equalities (17) and (18) link layered graph arcs to original arcs. Delay-variation-bound D is guaranteed by (19) and (20). In principle, variables x_{uv} and y^l_v are redundant since they can be substituted by Boolean layered graph arc variables x^k_{uv} using equalities (15), (17) and (18). However, model LAY is better readable by including them and branching on x_{uv} and Boolean y^l_v variables turned out to be more efficient in practice than branching on variables x^k_{uv}. In fact, branching on original arcs usually is more balanced since setting $x^l_{uv} = 1$ for one particular layered graph arc in general is a stronger constraint on the set of feasible solutions than setting $x_{uv} = 1$.

6 Valid Inequalities

The following sets of valid inequalities are not necessary for the feasibility of model LAY but are useful to strengthen it w.r.t. its linear programming (LP) relaxation denoted by LAY_{LP}.

6.1 Directed Connection Inequalities

The following constraints describe the well-known directed connection inequalities defined on original graph G':

$$\sum_{(u,v)\in A,\ u\in W,\ v\notin W} x_{uv} \geq 1 \qquad \forall W \subset V,\ s \in W,\ (V \setminus W) \cap R \neq \emptyset \tag{24}$$

Let LAY^{dc} denote the variant of model LAY with those inequalities included. Constraint (24) with $W = \{s\}$ ensures at least one arc going out of the root, and a subset of the subtour elimination constraints (equivalent to constraints (24)) with two-node-sets prevents cycles of length two:

$$\sum_{(s,v)\in A} x_{sv} \geq 1 \qquad \text{and} \qquad x_{uv} + x_{vu} \leq 1 \qquad \forall \{u, v\} \in E \tag{25}$$

Model LAY extended just by constraints (25) is denoted LAY^{r2}.

A stronger variant of (24) can be defined on layered graph G_{L}. For this purpose, we extend G_{L} by additional terminal nodes and arcs. Let $G'_{\mathrm{L}} = (V'_{\mathrm{L}}, A'_{\mathrm{L}})$

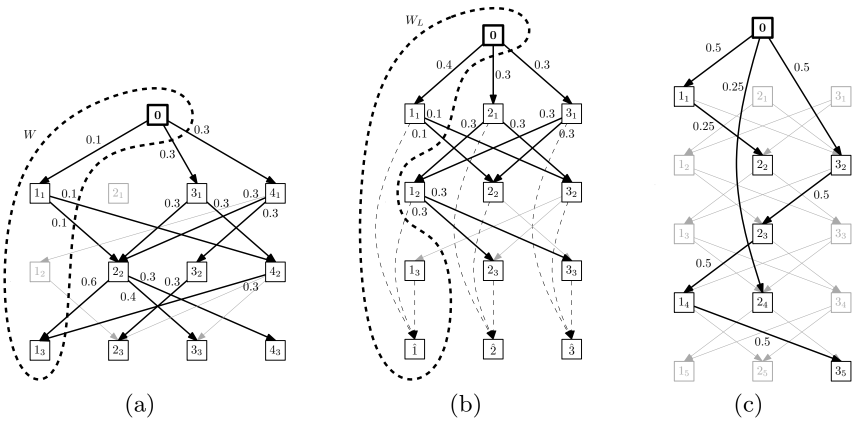

Fig. 3. All three examples are feasible for LAY_{LP} (arc labels denote variable values of the LP solution, gray arcs mean $x_{uv}^k = 0$). (a) The solution violates inequality (24) with $W = \{0, 1\}$ and root-constraint (25). (b) Inequality (26) with $W_{\mathrm{L}} = \{0, 1_1, 1_3, \hat{1}\}$ is violated. (c) The solution with $D = 1$ is feasible for constraints (27) but not for (28)–(30) since $y_1^1 + y_2^3 + y_2^4 = 1.25 > 1$ or $y_1^1 + y_2^3 + y_3^5 = 1.5 > 1$.

be the graph with nodes $V'_{\mathrm{L}} = V_{\mathrm{L}} \cup R_{\mathrm{L}},\ R_{\mathrm{L}} = \{\hat{v} \mid v \in R\}$ and arc set $A'_{\mathrm{L}} = A_{\mathrm{L}} \cup \hat{A},\ \hat{A} = \{(v_l, \hat{v}) \mid v_l \in V_{\mathrm{L}},\ \hat{v} \in R_{\mathrm{L}},\ v \in R\}$. We can now write

$$\sum_{(u_k,v_l) \in A'_{\mathrm{L}},\ u_k \in W_{\mathrm{L}},\ v_l \notin W_{\mathrm{L}}} x_{uv}^k \geq 1 \qquad \forall W_{\mathrm{L}} \subset V'_{\mathrm{L}},\ s \in W_{\mathrm{L}},\ (V'_{\mathrm{L}} \setminus W_{\mathrm{L}}) \cap R_{\mathrm{L}} \neq \emptyset. \tag{26}$$

We denote model LAY augmented by constraints (26) by LAY^{ldc}. It can be easily seen that inequalities (26) include (24). Fig. 3(a) and 3(b) show examples for strengthening model LAY_{LP}.

Inequalities (25) are included in the model a priori while (24) and (26) need to be separated dynamically during branch-and-cut. Violated inequalities are found via maximum flows (FIFO push-relabel method [2]) in a support graph using the current optimal LP relaxation values as arc capacities. Capacities for arcs $\hat{A}$ are set to 1.

6.2 Delay-Variation Inequalities

Let $L_v = \{l \mid v_l \in V_{\mathrm{L}}\} \subseteq \{1, ..., B\}$ denote the set of possible layers in G_{L} for a node $v \in V$. We know that a terminal node $u_k \in V_{\mathrm{L}},\ u \in R$, on layer $k \in L_u$ can only be in a feasible solution if no other terminal node $v_l \in V_{\mathrm{L}},\ v \in R$, on layer $l \in L_v$ outside the interval $[k - D, k + D]$ is included. This leads to inequalities

$$y_u^k + y_v^l \leq 1 \qquad \forall u, v \in R,\ \forall k \in L_u,\ \forall l \in L_v,\ |k - l| > D. \tag{27}$$

The number of inequalities (27) is in $\mathcal{O}(|R|^2 \cdot B^2)$. We can aggregate them to form stronger constraints

$$y_u^k + \sum_{l \in L_v \setminus \{k-D,\dots,k+D\}} y_v^l \leq 1 \qquad \forall u, v \in R,\ \forall k \in L_u. \tag{28}$$

The number of these is in $\mathcal{O}(|R|^2 \cdot B)$. Now we relate arbitrary subsets of layers of two terminal nodes leading to a violation of the delay-variation-constraint:

$$\sum_{l \in L'_u} y_u^l + \sum_{l \in L'_v} y_v^l \leq 1 \qquad \forall u, v \in R,\ \forall L'_u \subseteq L_u,\ \forall L'_v \subseteq L_v \text{ with}$$
$$|l_u - l_v| > D,\ \forall l_u \in L'_u,\ \forall l_v \in L'_v \tag{29}$$

In the most general variant we consider infeasible combinations of arbitrary subsets of layers of an arbitrary subset of terminal nodes:

$$\sum_{v \in R'} \sum_{l \in L'_v} y_v^l \leq 1 \qquad \forall R' \subseteq R,\ \forall v \in R',\ \forall L'_v \subseteq L_v, \text{ with}$$
$$|l_u - l_v| > D,\ \forall u, v \in R',\ \forall l_u \in L'_u,\ \forall l_v \in L'_v \tag{30}$$

Note that due to the inequalities' conditions w.r.t. R', v, and L'_v, the sum on the left side can include at most B y-variables, but the number of constraints can be exponential. We denote model LAY with constraints (30) by LAY^{dv}. In Fig. 3(c) an example is given where constraints (28)–(30) tighten LAY_{LP}.

To find violated inequalities (30) we consider an optimal LP solution S and build a support graph $G_S = (V_S, A_S)$ with node set $V_S = \{s\} \cup \{v_l \in V_{\text{L}} \mid v \in R,\ y_v^l > 0\}$ and arcs $A_S = \{(s, v_l) \mid v_l \in V_S \setminus \{s\}\} \cup \{(v_k, v_l) \mid v_k, v_l \in V_S \setminus \{s\},\ k < l,\ \nexists v_i \in V_S : k < i < l\} \cup \{(u_k, v_l) \mid u_k, v_l \in V_S \setminus \{s\},\ u \neq v,\ k < l,\ l - k > D\}$. Furthermore, we assign arc costs $c_a = y_v^l,\ \forall a = (u_k, v_l) \in A_S$.

Lemma 1. *Given an LP solution S and the corresponding graph G_S, each path $P \subseteq A_S$ with source node s and costs $c(P) > 1$ corresponds to an inequality* (30) *I_S^P by solution S and vice versa.*

Proof. Assume a path P in G_S starting in s with costs $c(P) > 1$ is given. By relating arc $a = (u_k, v_l)$ to variable y_v^l the sum of arc costs of P corresponds to a sum of y_v^l-variable values since $c_a = y_v^l$. Due to the definition of G_S P can only consist of arcs $(u_k, v_l) \in A_S$ with $k < l$ and either $u = v$ or $u \neq v \wedge l - k > D$. Therefore the sum of variables y_v^l corresponding to a path P forms the left side of a feasible inequality (30) and since $c(P) > 1$ we obtain a violated inequality I_S^P for solution S. Now, let I_S be a violated inequality (30) for solution S. First we remove all variables with $y_v^l = 0$ and sort the remaining sum of variables y_v^l by ascending layers l. Due to the constraint definition no two variables can have the same layer l and if we consider two consecutive variables y_u^k and y_v^l then either $u = v$ or $u \neq v \wedge l - k > D$. Furthermore, there has to be either an arc $(u_k, v_l) \in A_S$ or in case of $u = v$ possibly a path $P' = (u_k, \dots, u_l)$ including other nodes u_i with $k < i < l$. So the series of variables in I_S can again be related to a path P in G_S starting in s and since the sum of variable values is larger than 1 the costs of path P are at least that high. □

Following Lemma 1 we now search for the longest paths from s to at most $|R|$ leaves in G_S. The single-source longest path problem can here be solved in linear time since G_S is a directed acyclic graph, cf. [1]. Obviously, all inequalities $I_S^{P'}$ corresponding to sub-paths $P' \subset P$ with $c(P') > 1$ are dominated by I_S^P. To further strengthen inequality I_S^P we try to feasibly add as many summands as possible, not only the node variables which are positive in solution S. Otherwise similar violated inequalities are possibly found in further iterations. So if we consider an arc $(v_k, v_l) \in A_S$ on a violating path P connecting two layered graph nodes corresponding to the same original node we additionally add all variables $y_v^i, \ \forall v_i \in V_{\mathrm{L}}, \ k < i < l$, to I_S^P. Using this separation routine we are able to guarantee that the "most violated" inequalities are found hopefully resulting in a large increase of the optimal LP relaxation value.

7 Experimental Results

We implemented all models using IBM CPLEX 12.3 as MIP solver with default settings. Each run has been performed on a single core of an Intel Xeon E5540 processor with 2.53 GHz, and an absolute limit of 10 000 CPU-seconds has been applied to each experiment. We tested our models on instances originally proposed by Gouveia et al. [4] for the spanning tree variant of the RDCST problem, focusing on the most difficult subset E with Euclidean costs and the root s placed near the border. Each instance set consists of five complete input graphs with 21 or 41 nodes and a specific range of possible discrete edge delay values, e.g. E21-10 denotes the set of instances where $|V| = 21$ and $d_e \in \{1, \ldots, 10\}$, $\forall e \in E$. We set $R = \{0, \ldots, \lfloor |V|/2 \rfloor\}$, $D \in \{1, 3\}$ for sets E21-10 and E41-10, and $D \in \{10, 30\}$ for set E21-100. We applied all mentioned preprocessing methods for graph reduction prior to solving. It turned out to be beneficial to declare flow variables f_{uv}^w and layered graph variables y_v^l and x_{uv}^k integer since CPLEX can make use of it both to reduce the model size in the presolving phase and to speed up the solving process by additionally branching on these variables.

Test results comparing different model variants are shown in Table 1 where dashes denote either a 100% gap or reached time limit. In general, the obtained integrality gaps even if adding all valid inequalities, are much higher than those of the corresponding RDCST problem without the delay-variation-constraint, cf. [14]. This documents that the delay-variation-constraint indeed imposes a big additional challenge. It can clearly be seen that while the LAY variants performed at least reasonably well, model MCF' is not competitive in most cases. Only for the small E21-100 instances MCF' can sometimes outperform the other methods since here the number of nodes is quite low resulting in a manageable number of flow variables and the delay-bounds are rather high which is disadvantageous for the layered graph approaches.

Note that none of the polyhedrons of MCF'_{LP} and LAY_{LP} without additional valid inequalities dominates the other since there are cases where the optimal LP values of the first model are better than those of the second and vice versa. In case of small delay-bounds layered graph models mostly outperform other

Table 1. Comparison of models (1: MCF', 2: LAY^{r2}, 3: $LAY^{\mathrm{r2/dc}}$, 4: $LAY^{\mathrm{r2/ldc}}$, 5: $LAY^{\mathrm{r2/dv}}$, 6: $LAY^{\mathrm{r2/dv/dc}}$, 7: $LAY^{\mathrm{r2/dv/ldc}}$) on test sets from [4] (B: delay-bound, D: delay-variation, #opt: number of optimal solutions (out of 5), $\overline{\text{gap}}$: average gap in percent, t: median CPU time in seconds; best results are printed bold)

			#opt							$\overline{\text{gap}}$ [%]							t [s]						
Set	B	D	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
E21-10	10	1	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	2599	**1**	**1**	**1**	**1**	**1**	**1**
		3	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	1882	**2**	**2**	4	**2**	**2**	**2**
	15	1	0	**5**	**5**	**5**	**5**	**5**	**5**	23.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	-	**6**	13	126	13	11	73
		3	1	**5**	**5**	**5**	**5**	**5**	**5**	14.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	-	18	25	346	**16**	21	117
	20	1	0	**5**	**5**	2	**5**	**5**	3	31.2	**0.0**	**0.0**	7.3	**0.0**	**0.0**	4.1	-	**55**	85	-	210	181	1077
		3	0	**5**	**5**	1	**5**	**5**	4	18.9	**0.0**	**0.0**	14.8	**0.0**	**0.0**	2.3	-	284	870	-	**220**	243	7381
	25	1	0	**5**	**5**	0	4	3	1	27.9	**0.0**	**0.0**	19.1	1.5	4.8	14.2	-	**347**	434	-	4592	2330	-
		3	0	**5**	**5**	1	**5**	**5**	1	17.2	**0.0**	**0.0**	46.2	**0.0**	**0.0**	12.3	-	**787**	1730	-	2934	3115	-
E41-10	10	1	0	**5**	**5**	**5**	**5**	**5**	**5**	-	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	-	**26**	43	414	**26**	33	79
		3	0	**5**	**5**	4	**5**	**5**	**5**	62.1	**0.0**	**0.0**	0.5	**0.0**	**0.0**	**0.0**	-	116	191	779	**43**	63	223
	15	1	0	**4**	0	0	1	0	0	-	**2.7**	15.6	24.3	8.7	10.3	24.4	-	**3631**	-	-	-	-	-
		3	0	2	0	0	**4**	2	0	-	8.5	15.1	30.2	**0.8**	4.3	17.2	-	-	-	-	**6707**	-	-
	20	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	-	27.5	**26.1**	88.2	34.9	32.7	56.0	-	-	-	-	-	-	-
		3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	-	28.3	27.6	71.2	22.8	**18.8**	51.4	-	-	-	-	-	-	-
	25	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	-	35.3	**35.2**	-	52.5	68.4	-	-	-	-	-	-	-	-
		3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	-	35.3	33.9	-	27.6	**26.8**	71.1	-	-	-	-	-	-	-
E21-100	100	10	0	**5**	**5**	4	**5**	**5**	**5**	19.8	**0.0**	**0.0**	0.8	**0.0**	**0.0**	**0.0**	-	**42**	76	1489	46	49	168
		30	4	**5**	**5**	**5**	**5**	**5**	**5**	1.7	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	2733	89	193	349	79	**36**	65
	150	10	0	**4**	2	0	1	0	0	45.5	**2.9**	10.0	53.9	23.6	20.4	67.9	-	**2152**	-	-	-	-	-
		30	1	**5**	4	1	4	**5**	1	13.9	**0.0**	0.6	27.1	1.6	**0.0**	8.6	-	3405	**2034**	-	8009	7673	-
	200	10	0	**1**	0	0	0	0	0	37.3	**22.5**	33.5	91.0	59.9	76.4	-	-	-	-	-	-	-	-
		30	**2**	1	1	0	1	1	1	**8.4**	17.2	15.8	46.0	17.5	17.8	63.2	-	-	-	-	-	-	-
	250	10	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**48.0**	62.6	66.0	-	-	-	-	-	-	-	-	-	-	-
		30	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**22.0**	26.0	26.0	80.0	37.6	37.3	80.0	-	-	-	-	-	-	-

modeling approaches, cf. [4,5,14]. However, we can also notice the disadvantage of such models that increasing delay-bounds result in higher runtimes. When comparing different layered graph models, LAY^{r2} and $LAY^{\mathrm{r2/dv}}$ performed best. Obviously, directed connection cuts are rarely in graph G' and never in G_{L} helpful to improve computation times. Reasons for this are both the higher complexity of the separation problem compared to the fast method for finding violated inequalities (30) and the fact that in most cases the number of added connection cuts is rather high leading to slow LP relaxation computations.

8 Conclusion and Future Work

We tackled the rooted delay- and delay-variation-constrained Steiner tree problem by using two different MIP models based on multi-commodity-flows and a layered graph transformation. Furthermore, we proposed sets of valid inequalities for the second model particularly targeting the bounding of the delay-variation and provided an efficient separation method. Experimental results clearly show the superiority of layered graph models with or without delay-variation cuts. Nevertheless, the generally still relatively large integrality gaps of the LAY models ask for investigating also other modeling approaches, e.g. path models. New insights can be achieved by comparing the polyhedra of our models in detail, possibly leading to further strengthening inequalities. For addressing the poor scalability of the LAY models w.r.t. larger delay-bounds, an appropriate extension of the adaptive layers framework [14] seems to be highly promising.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications. Prentice Hall (1993)
2. Cherkassky, B.V., Goldberg, A.V.: On Implementing the Push-Relabel Method for the Maximum Flow Problem. Algorithmica 19(4), 390–410 (1997)
3. Gouveia, L.: Multicommodity flow models for spanning trees with hop constraints. European Journal of Operational Research 95(1), 178–190 (1996)
4. 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)
5. Gouveia, L., Simonetti, L.G., Uchoa, E.: Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. Mathematical Programming 128(1), 123–148 (2011)
6. Haberman, B.K., Rouskas, G.N.: Cost, delay, and delay variation conscious multicast routing. Tech. rep., North Carolina State University (1996)
7. Koch, T., Martin, A.: Solving Steiner tree problems in graphs to optimality. Networks 32(3), 207–232 (1998)
8. Kun, Z., Heng, W., Feng-yu, L.: Distributed multicast routing for delay and delay variation-bounded Steiner tree using simulated annealing. Computer Communications 28(11), 1356–1370 (2005)
9. Lee, H.-Y., Youn, C.-H.: Scalable multicast routing algorithm for delay-variation constrained minimum-cost tree. In: IEEE International Conference on Communications, vol. 3, pp. 1343–1347. IEEE Press (2000)
10. Ljubic, 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), 427–449 (2006)
11. Low, C.P., Lee, Y.J.: Distributed multicast routing, with end-to-end delay and delay variation constraints. Computer Communications 23(9), 848–862 (2000)
12. Rouskas, G.N., Baldine, I.: Multicast routing with end-to-end delay and delay variation constraints. IEEE Journal on Selected Areas in Communications 15(3), 346–356 (1997)
13. Ruthmair, M., Raidl, G.R.: Variable Neighborhood Search and Ant Colony Optimization for the Rooted Delay-Constrained Minimum Spanning Tree Problem. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI, Part II. LNCS, vol. 6239, pp. 391–400. Springer, Heidelberg (2010)
14. Ruthmair, M., Raidl, G.R.: A Layered Graph Model and an Adaptive Layers Framework to Solve Delay-Constrained Minimum Tree Problems. In: Günlük, O., Woeginger, G.J. (eds.) IPCO 2011. LNCS, vol. 6655, pp. 376–388. Springer, Heidelberg (2011)
15. Sheu, P.-R., Chen, S.-T.: A fast and efficient heuristic algorithm for the delay- and delay variation-bounded multicast tree problem. Computer Communications 25(8), 825–833 (2002)
16. Sheu, P.-R., Tsai, H.-Y., Chen, S.-C.: An Optimal MILP Formulation for the Delay- and Delay Variation-Bounded Multicast Tree Problem. Journal of Internet Technology 8(3), 321–328 (2007)