Common-Flow Formulations for the Diameter Constrained Spanning and Steiner Tree Problems

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

[1]Universidade de Lisboa, Faculdade de Ciências, Departamento de Estatística e Investigação Operacional, Lisbon, Portugal. legouveia@fc.ul.pt
[2]Department of Operations Analytics, Vrije Universiteit Amsterdam, Netherlands. m.leitner@vu.nl
[3]ESSEC Business School of Paris, France. ljubic@essec.edu

April 28, 2023

Abstract

We consider the diameter constrained minimum Steiner tree problem on a graph (DCStTP). Given an edge-weighted undirected graph whose set of nodes is partitioned into a set of terminal and potential Steiner nodes, the objective is to find a minimum-weight subtree that spans all terminal nodes such that the number of hops between any two terminal nodes does not exceed a given diameter D. In this work, we introduce mixed-integer linear programming models for the DCStTP based on the concept of triangles, i.e. diameter constrained Steiner trees induced by terminal subsets of size three. Starting from a formulation that models a D-hop Steiner arborescence rooted at a randomly chosen terminal node, we discuss various possibilities of realizing triangles using multi-commodity, common, or uncommon flows. We analyse the strength of these models both theoretically and empirically, and investigate how the respective Benders reformulations influence the computational performance.

1 Introduction

We consider the diameter constrained variant of the Steiner tree problem on a graph (DCStTP). Formally, we are given an undirected graph $G = (V, E)$, with node set V, edge set E, edge costs $c_e \geq 0$, for all $e \in E$, and a diameter limit $D \in \mathbb{N}, 2 \leq D \leq |V| - 1$. The node set V is the disjoint union of terminal nodes T and potential Steiner nodes S. A feasible solution to the DCStTP is a subtree $G_S = (V_S, E_S)$ connecting all terminal nodes (also known as a Steiner tree, where $T \subset V_S$), whose diameter does not exceed D. Hence, the maximum number of edges between any pair of nodes in that tree is bounded from above by the value of D. For $D \in \{2, 3\}$, the problem can be solved in polynomial time. For $4 \leq D \leq |V| - 2$, the problem is NP-hard, even if we consider the spanning-tree variant of the problem (namely, $S = \emptyset$ and $T = V$), see Garey and Johnson (1979). The latter variant is known as the Diameter Constrained Spanning Tree problem on a graph, and we will refer to it as DCSpTP.

There exist important applications of the DCStTP in telecommunications, data compression, or parallel computing (see, e.g., Bookstein and Klein (1991); Deo and Abdalla (2000); Santos et al. (2016)). In telecommunication networks, for example, the latency of a signal sent between a pair of nodes is directly proportional to the number of edges of the routing path. In networks

that employ the so-called tree-multicasting, the maximum latency thus corresponds to the diameter of that tree (Vik et al., 2008).

Given a tree $G_S = (V_S, E_S)$, the *eccentricity* of a node v is the maximum number of edges on a path from v to any other node in G_S. The diameter of G_S is thus the maximum eccentricity over all nodes $v \in V_S$. Given a diameter D, the *center* of G_S is a node (if D is even) or a pair of adjacent nodes (if D is odd) of minimum eccentricity. Thus, a DCStTP can be seen as a Steiner tree rooted at an unknown center whose eccentricity (i.e., maximum number of hops to another node from G_S) H is bounded by the half of the diameter, i.e. $H = \lfloor D/2 \rfloor$. Many flow-based formulations have been proposed based on these concepts related to centers of trees and hop constraints. One of the first compact formulations for the DCStTP has been proposed by Achuthan et al. (1994). It is based on the Miller-Tucker-Zemlin (MTZ) modeling approach for trees, and it has been later improved by dos Santos et al. (2004). Formulations based on multi-commodity flows have been later studied by Gouveia and Magnanti (2003); Gouveia et al. (2004, 2006). Compared to MTZ-based formulations, the multi-commodity-flow models typically provide much tighter linear programming (LP) bounds at the expense of involving a much larger number of flow variables. Most of these compact models are not (directly) suitable to be used for solving the DCStTP on very large graphs (either because of very weak lower bounds, or due to the prohibitive size of the underlying formulations). The most successful method for solving the DCSpTP is based on a layered graph reformulation in which the original problem is restated as a Steiner arborescence problem with additional constraints on a very large graph. Copies of the original nodes are created at each layer and the problem is solved using a branch-and-cut procedure (Gouveia et al., 2011). Most of the existing studies focus on the DCSpTP problem variant (notable exceptions are studies by Gouveia and Magnanti (2003); Gouveia et al. (2004, 2006)), hence slight model modifications are needed in order to address the more general problem variant, namely the DCStTP. The question on how to provide a valid formulation for the DCSpTP in the *natural space* of edge variables (using only one variable associated to each edge) remained open for some time. Gouveia et al. (2020) were the first to provide a valid model in the space of edge-variables using the concept of *circular jump* constraints. The authors provide a polyhedral study of the DCSpTP polytope and show that most of the proposed inequalities are not implied by the currently best-performing extended formulation based on layered graphs (Gouveia et al., 2011).

Heuristics have been investigated in Raidl and Julstrom (2003); Singh and Gupta (2007); Gruber et al. (2006); Binh et al. (2009); Lucena et al. (2010); Requejo and Santos (2009); Steitz (2015); Santos et al. (2016), and a constraint programming approach has been proposed by Noronha et al. (2010). Another problem variant, called the two-level diameter constrained spanning tree problem, has been introduced by Gouveia et al. (2015). For a recent survey on the Steiner tree problem in graphs (STP), see Ljubić (2021).

Our contribution: In this work, we introduce three mixed-integer linear programming (MILP) models for the DCStTP. We start with a MILP formulation that models a D-hop Steiner arborescence rooted at a randomly chosen terminal node. In order to derive valid formulations for the DCStTP, we then exploit the concept of triangles, i.e. we model diameter constrained Steiner trees induced by terminal subsets of size three. The first approach ensures that pairwise connections between any two terminals different than the root are not longer than D (cf. Section 2.1). For the second model, we start from the concept of common flows introduced by Polzin and Daneshmand (2001) to strengthen the multi-commodity flow formulation for Steiner trees. In the context of the DCStTP, we show that common flows are not only useful for strengthening the bounds, but provide an alternative way to impose diameter constraints (cf. Section 3.1). Finally, our third model is based on a new concept, to which we refer as *uncommon flow* (cf.

Section 3.2). We analyse the strength of these models both theoretically and empirically, and investigate how the respective Benders reformulations influence the computational performance (cf. Section 4).

This article is dedicated to Bernard Gendron whose scientific work has laid important foundations for many network optimization problems. Bernard made invaluable contributions in studying and improving standard *multi-commodity flow formulations* for various network design, location and routing problems (see, e.g., Gendron and Semet (2009); Contardo et al. (2013); Thiongane et al. (2015); Chouman et al. (2017)). Moreover, together with his co-authors, in his seminal works (Croxton et al., 2007; Gendron and Gouveia, 2017) Bernard has shown how *disaggregation* techniques can be applied to solve network flow problems with piecewise linear costs. The concepts of common and uncommon flows studied in our paper can be seen as an application of disaggregation techniques to multi-commodity flows, one of Bernard's favorite topics (Frangioni and Gendron, 2009).

2 Standard flow-based model

We will address directed models, that is, we will view the Steiner tree as a subtree directed away from a node in T, e.g, node 0. To define a directed tree we need to "direct" the graph, that is, for each edge $\{i, j\}$ we also distinguish whether the edge is used in the direction from i to j (arc (i, j) is used) or is used the direction from j to i (arc (j, i) is used) and we denote by A the set of arcs defined in this way. That way, we search for a Steiner arborescence directed away from 0, with a subset of remaining terminals as leaf nodes, and such that the diameter of the undirected counterpart of this arborescence does not exceed D.

In this section we start with a hop-constrained relaxation of the problem, based on the trivial property that if a tree has diameter D then the path between any given node (say 0) chosen as a root and any other terminal $t \in T \setminus 0$, has no more than D edges (or, in the case that we consider directed paths, it has no more than D arcs). Hence, any minimum-cost D-hop constrained Steiner arborescence rooted at 0, spanning all terminals from T, provides a valid lower bound to our problem. Indeed, let $\mathcal{T}$ be a subtree of G that spans all the terminals. If we fix a terminal node $0 \in T$ as a root node of the arborescence associated with $\mathcal{T}$, then the maximum number of hops between 0 and any other node in $\mathcal{T}$ is at most D. This property however does not guarantee that the diameter of $\mathcal{T}$ is at most D, if the condition is imposed *only* for a fixed node 0 (indeed, the diameter can be as large as $2D$). However, if we impose this property to hold for any choice of $0 \in T$, then it is not difficult to see that the resulting tree will have a diameter not larger than D.

In this article, we start from the multi-commodity flow model for the D-hop constrained Steiner arborescence, and we propose three different ways to extend this model so as to ensure that the diameter of the arborescence does not exceed D. The first model (see Section 2.1) is a formulation in which we model paths between any two pairs of terminals, the remaining two models (see Sections 3.1 and 3.2) are based on the concepts of the common or uncommon flow, respectively.

To model the D-hop constrained problem, we consider the binary arc variables a_{ij}, for all $(i, j) \in A$, that indicate whether arc (i, j) is in a solution and the directed binary path variables f^t_{ij}, for all $t \in T \setminus \{0\}$ and all $(i, j) \in A$, indicating whether arc (i, j) is in the path from 0 to t. The model, denoted by (D-hop), reads as follows:

$$(D\text{−hop}) \ \min \sum_{(i,j)\in A} c_{ij} a_{ij} \tag{1a}$$

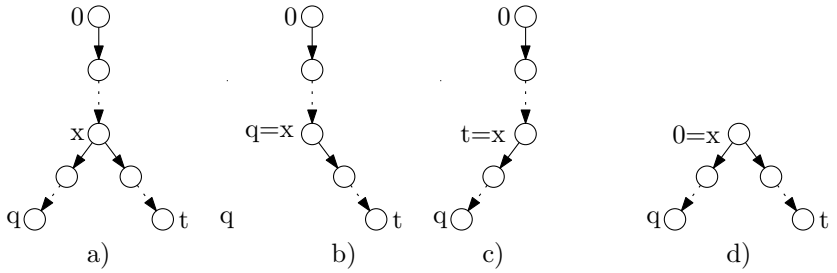

Figure 1: Feasible triangle graphs for $(0, q, t)$. a) standard case, b)-d) special cases where one terminal node is the splitting node x. In either case $l_{0q} \leq D$, $l_{0t} \leq D$, and $l_{xq} + l_{xt} \leq D$ must hold, where l_{ij} denotes the number of hops in a path from i to j.

$$\sum_{(j,i)\in A} a_{ji} \begin{cases} = 1 & \text{if } i \in T \setminus \{0\} \\ = 0 & \text{if } i = 0 \\ \leq 1 & \text{otherwise} \end{cases} \qquad i \in V \tag{1b}$$

$$\sum_{(i,j)\in A} a_{ij} \geq \sum_{(j,i)\in A} a_{ji} \qquad i \in S \tag{1c}$$

$$\sum_{(i,j)\in A} f_{ij}^t - \sum_{(j,i)\in A} f_{ji}^t = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{if } i \neq 0, t \\ -1 & \text{if } i = t \end{cases} \qquad t \in T \setminus \{0\},\ i \in V \tag{1d}$$

$$\sum_{(i,j)\in A} f_{ij}^t \leq D \qquad t \in T \setminus \{0\} \tag{1e}$$

$$0 \leq f_{ij}^t \leq a_{ij} \qquad t \in T \setminus \{0\},\ (i,j) \in A \tag{1f}$$

$$a_{ij} \in \{0, 1\} \qquad \forall (i,j) \in A \tag{1g}$$

Constraints (1b) are in-degree constraints, guaranteeing that the in-degree of each terminal node is one, except for the root node 0. The in-degree of each potential Steiner node is at most one. Inequalities (1c) state that for each potential Steiner node, the out-degree is greater than or equal to its in-degree. Constraints (1d) ensure that one unit of flow is sent from 0 to every other terminal node, and the capacity constraints (1f) ensure that the flow can be sent only along the arcs of the arborescence. Finally, inequalities (1e) guarantee that the number of arcs in any 0-t path is not larger than D. Also, a well known fact is that in the context of this model the flow variables can be defined as continuous, that is

$$f_{ij}^t \geq 0, \quad \forall t \in T \setminus \{0\}, \quad \forall (i,j) \in A. \tag{2}$$

Hence, the optimal solution of model (1) is a D-hop constrained minimum Steiner arborescence.

2.1 Flow-Based Model Based on Pairwise Terminal Connections

We now extend model ($D-$hop) using additional flow variables, denoted as g. The value of g_{ij}^{qt} ($q, t \in T \setminus \{0\}$, $q < t$, $(i,j) \in A$) is equal to one if any only if we can route one unit of flow from q to t along the edge $\{i,j\}$ (notice that we need to "ignore" the direction of the arc, as explained below). Hence, to ensure that the diameter of the arboresce whose incidence vector is given by a is not larger than D, the length of any q-t path should not exceed D. The resulting model, that we will denote by (SF), reads as follows:

$$(SF)\ \min \sum_{(i,j)\in A} c_{ij}a_{ij}$$

$$\sum_{(i,j)\in A} g_{ij}^{qt} - \sum_{(j,i)\in A} g_{ji}^{qt} = \begin{cases} 1 & \text{if } i = q \\ 0 & \text{if } i \neq q,t \\ -1 & \text{if } i = t \end{cases} \qquad q,t \in T\setminus\{0\},\ q<t,\ i \in V \tag{3a}$$

$$\sum_{(i,j)\in A} g_{ij}^{qt} \leq D \qquad q,t \in T\setminus\{0\},\ q<t \tag{3b}$$

$$0 \leq g_{ij}^{qt} + g_{ji}^{qt} \leq a_{ji} + a_{ij} \qquad q,t \in T\setminus\{0\},\ q<t,\ \{i,j\} \in E \tag{3c}$$

$$(a,f) \text{ satisfy (1b)-(1g)}$$

Constraints (3a) guarantee that one unit of flow is sent from each $q \in T$ to each $t \in T$, $q < t$. Inequalities (3b) ensure that each q-t path is of length at most D, and finally the capacity constraints (3c) impose that the flow g can only be sent along the arcs of the Steiner arborescence determined by a. We note that the direction of arcs and the direction of the flow g do not need to be identical. Consider an example given in Figure 1(a) which demonstrates a Steiner arborescence rooted at 0 with terminals q and t. In this solution, the flow g^{qt} will use arcs a_{ij} in the opposite direction, i.e., $g_{ij}^{qt} = a_{ji}$ for all arcs between node x (the least common ancestor between q and t) and q, whereas $g_{ij}^{qt} = a_{ij}$ for all arcs connecting node x with t. This is why the capacities given on the right-hand-side have to be considered in an undirected fashion.

One may additionally consider constraints

$$0 \leq g_{ij}^{qt} \leq f_{ji}^{q} + f_{ij}^{t} \qquad q,t \in T\setminus\{0\},\ q<t,\ (i,j) \in A \tag{4}$$

stating that the flow g can only be sent along arcs determined by the flow f. Since q is the destination node for f and a source node for g, if arc (i,j) is used along the q-t path, this arc corresponds to the opposite direction of the flow f^q. Similarly, since t is the destination node for both f and g, the orientation of the arc (i,j) will be identical for both of them.

The LP relaxation of the previous model can be further enhanced by adding the following *bidirectional linking* constraints, motivated by Balakrishnan et al. (1989):

$$g_{ij}^{kt} + g_{ji}^{qt} \leq a_{ij} + a_{ji} \qquad k,q,t \in T\setminus\{0\}, k<t, q<t, k\neq q, \{i,j\} \in E \tag{5a}$$

$$g_{ij}^{kq} + g_{ji}^{kt} \leq a_{ij} + a_{ji} \qquad k,q,t \in T\setminus\{0\}, k<q, k<t, q\neq t, \{i,j\} \in E \tag{5b}$$

$$g_{ji}^{kq} + g_{ij}^{kt} \geq g_{ij}^{qt} \qquad k,q,t \in T\setminus\{0\}, q<t, k<q, \{i,j\} \in E \tag{5c}$$

The first set of constraints states that we cannot have flow in opposite directions of a given edge going from a common origin to two different destinations. The second set has a similar symmetrical explanation. The third set "triangulates" the flow for a given pair of nodes in terms of a third node, that is, if arc (i,j) is in the path from q to t, then for any other node k, either it is in the path from k to t, or the reversed arc, (j,i), should be in the path from k to q.

The LP relaxation of this standard model can be further strengthened by adding the following set of constraints

$$f_{ij}^{q} + f_{ij}^{t} + f_{ji}^{q} + f_{ji}^{t} + g_{ij}^{qt} + g_{ji}^{qt} \leq 2(a_{ij} + a_{ji}) \qquad q,t \in T\setminus\{0\}, q<t, \{i,j\} \in E \tag{6}$$

The validity of these constraints follows from the fact that the left hand-side of (6) cannot be greater than two. Observe that for a given pair $(i,j) \in A$, these constraints are a stronger

version of an inequality that is obtained by adding (3c) for the same pair and four constraints (1f), respectively for the pairs (i, j) and (j, i) and destinations q and t. We will link constraints (6) with the uncommon flow formulation in Section 3.2.

Let us denote the models considered so far by (SF), (SF)+(4), (SF)+(4)-(5) and (SF)+(4) -(6). Let $v_{\text{LP}}(F)$ denote the value of the LP-relaxation of a given formulation F. We can derive the following result:

Proposition 1. *We have:*

$$v_{\text{LP}}(SF) \leq v_{\text{LP}}((SF) + (4)) \leq v_{\text{LP}}((SF) + (4) - (5)) \leq v_{\text{LP}}((SF) + (4) - (6)).$$

Moreover, there exist DCStTP instances for which the strict inequalities hold.

Instead of providing a proof of the second part of this result, we refer to our computational section where we provide some benchmark instances from the literature for which the strict inequalities between the proposed models hold.

3 Triangle-based Models

The model discussed in the previous section is based on a characterization stating that a tree $\mathcal{T}$ has diameter at most D if any Steiner tree with two terminals contained in $\mathcal{T}$ has diameter at most D. In this section we extend the characterization by stating that a tree $\mathcal{T}$ has diameter at most D if any Steiner tree with three terminals contained in $\mathcal{T}$ has diameter at most D. This generalization follows the idea of strengthening a standard multi-commodity-flow-based model using common flows, as proposed by Polzin and Daneshmand (2001) for Steiner trees.

To see this connection, note that the standard multi-commodity flow models are characterized by including a set of variables and a system of constraints associated to each (or some of the) pair of nodes, essentially guaranteeing that there is path connecting the pair of nodes for which the systems are defined. The previous formulation exemplifies one such model. For each pair of nodes, the model includes sets of variables, either f and g, and the corresponding systems. Polzin and Daneshmand (2001) have proposed a formulation that generalizes this concept by also associating variables to triples of nodes $(0, q, t)$ that model the flow that is common to the paths from node 0 to node q and from node 0 to node t. The main idea of using these new variables is that by also including a new set of coupling constraints (relating new flow variables with standard flow variables and arc design variables) is to decrease the possibility of the standard flow from 0 to q and from 0 to t to split up and then to rejoin again (something that often happens in LP solutions of standard flow models even in problems where the constraints do not allow cycles in the integer feasible solutions, such as trees).

In this work we provide another modeling possibility of using the common flow variables, namely allowing the derivation of the diameter constraints for each pair of nodes $q, t \in T \setminus \{0\}, q < t$. In fact, as we shall shown next, for obtaining a valid model for the DCStTP we did not even need to add flow balancing constraints for the new flow variables. Figure 1 illustrates the four potential realizations of a triangle contained in $\mathcal{T}$ which is rooted at $0 \in T$ and also contains terminals $q, t \in T$. Next, to the standard case shown in Figure 1(a), any of the three terminals (i.e., $0, q$ or t) may be the splitting node x until which the paths from 0 to q and from 0 to t separate, see Figures 1(b), (c), and (d).

3.1 A Common Flow Realization

In the common flow variant, cf. Polzin and Daneshmand (2001), each triangle $(0, q, t)$, $\forall q, t \in T \setminus \{0\}$, $q < t$, is modeled using flow variables $f_{ij}^q \geq 0$, $f_{ij}^t \geq 0$, introduced in the model (D-hop), and additional *common flow* variables $m_{ij}^{qt} \geq 0$ for each $(i, j) \in A$. The latter describes

the common part of flows f^q and f^t. The following model, denoted by (CF), provides a valid formulation for the DCStTP:

$$
\begin{aligned}
\text{(CF)}\ \min\ & \sum_{(i,j)\in A} c_{ij}a_{ij} \\
& m_{ij}^{qt} \le f_{ij}^{q} && q,t \in T\setminus\{0\},\ q<t,\ (i,j)\in A && \text{(7a)}\\
& m_{ij}^{qt} \le f_{ij}^{t} && q,t \in T\setminus\{0\},\ q<t,\ (i,j)\in A && \text{(7b)}\\
& f_{ij}^{q} + f_{ij}^{t} \le a_{ij} + m_{ij}^{qt} && q,t \in T\setminus\{0\},\ q<t,\ (i,j)\in A && \text{(7c)}\\
& \sum_{(i,j)\in A} (f_{ij}^{q} + f_{ij}^{t} - 2m_{ij}^{qt}) \le D && q,t \in T\setminus\{0\},\ q<t && \text{(7d)}\\
& (a,f) \text{ satisfy (1b)-(1g)}
\end{aligned}
$$

For each triple $(0,q,t)$, constraints (7a), (7b), and (7c) ensure that the common flow m_{ij}^{qt} on arc $(i,j) \in A$ is equal to one if both f_{ij}^{q} and f_{ij}^{t} are set to one. Constraints (7c) are the previously mentioned constraints that characterize the formulation of Polzin and Daneshmand (2001). These constraints allow us to write the diameter constraints without resorting to using the g flow. By restricting the number of arcs with flow f^q or f^t that are not part of their common flow constraints, inequalities (7d) ensure that the distance between q and t does not exceed D. Finally, observe that without these constraints, we would have a formulation for the D-hop constrained problem with a LP relaxation as tight as the LP relaxation of the formulation (D-hop).

Although not needed to obtain a valid formulation, we can still follow Polzin and Daneshmand (2001) and add, for each pair $q < t$, $q,t \in T\setminus\{0\}$, the additional flow conservation constraints for common flows m^{qt}:

$$
\sum_{(i,j)\in A} m_{ij}^{qt} - \sum_{(j,i)\in A} m_{ji}^{qt} \le \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{if } i \neq 0 \end{cases} \qquad q,t \in T\setminus\{0\},\ q<t,\ i \in V \qquad \text{(8)}
$$

We use (CF$^+$) to refer to model (CF)+(8). Polzin and Daneshmand (2001) show that their common flow model for the STP depends on the choice of the root node. Our computational experiments confirm that this is also the case for our (CF$^+$) model for the DCStTP.

3.2 An Uncommon Flow Realization

As observed previously, to ensure the diameter constraints between q and t, $q < t$, $q,t \in T\setminus\{0\}$, we need to restrict the total flow along those arcs on which either flow f^q or f^t, but not both of them, are used. Hence, we propose to use flow variables $u_{ij}^{qt} \ge 0$ modeling the *uncommon flow* of q and t.

$$
\begin{aligned}
\text{(UF)}\ \min\ & \sum_{(i,j)\in A} c_{ij}a_{ij} \\
& u_{ij}^{qt} \ge f_{ij}^{q} - f_{ij}^{t} && q,t \in T\setminus\{0\},\ q<t,\ (i,j)\in A && \text{(9a)}\\
& u_{ij}^{qt} \ge f_{ij}^{t} - f_{ij}^{q} && q,t \in T\setminus\{0\},\ q<t,\ (i,j)\in A && \text{(9b)}\\
& \sum_{(i,j)\in A} u_{ij}^{qt} \le D && q,t \in T\setminus\{0\},\ q<t && \text{(9c)}
\end{aligned}
$$

$$u_{ij}^{qt} \leq f_{ij}^{q} + f_{ij}^{t} \qquad q,t \in T \setminus \{0\},\ q < t,\ (i,j) \in A \tag{9d}$$

$$u_{ij}^{qt} \leq 2a_{ij} - f_{ij}^{q} - f_{ij}^{t} \qquad q,t \in T \setminus \{0\},\ q < t,\ (i,j) \in A \tag{9e}$$

(a, f) satisfy (1b)-(1g)

For each triple $(0, q, t)$, constraints (9a), (9b), (9d) and (9e) ensure that the uncommon flow u_{ij}^{qt} on arc $(i,j) \in A$ is equal to one if exactly one of values f_{ij}^{q} and f_{ij}^{t} is equal to one. By restricting the number of arcs with uncommon flow u^{qt} using constraints (9c), we ensure that the distance between q and t does not exceed D.

Proposition 2. *The models (CF) and (UF) are equally strong in terms of their LP-relaxations, i.e., for all instances of the DCStTP, we have:*

$$v_{\text{LP}}(\text{CF}) = v_{\text{LP}}(\text{UF}).$$

Proof. By using the following linear transformation

$$u_{ij}^{qt} = f_{ij}^{q} + f_{ij}^{t} - 2m_{ij}^{qt}, \qquad (i,j) \in A, q, t \in T, q < t$$

we can transform any feasible LP-solution of the model (CF) into an LP-feasible solution of the model (UF), and vice versa. Under this equality, constraint (9d) corresponds to $m_{ij}^{qt} \geq 0$. Other equivalences are: (9a) corresponds to (7a) and (9b) corresponds to (7b); (9e) corresponds to (7c); (9c) corresponds to (7d). Finally, note that an inequality in the common-flow model that corresponds to $u_{ij}^{qt} \geq 0$, is the inequality that results from adding (7a) with (7b) for the same indices. □

Additional flow conservation constraints for the uncommon flow can also be imposed:

$$\sum_{(i,j)\in A} u_{ij}^{qt} - \sum_{(j,i)\in A} u_{ji}^{qt} \geq \begin{cases} -1 & \text{if } i \in \{q,t\} \\ 0 & \text{otherwise} \end{cases} \qquad q,t \in T \setminus \{0\},\ q < t,\ \forall i \in V \tag{10}$$

We use (UF$^+$) to refer to the model (UF)+(10).

Proposition 3. *The models (CF$^+$) and (UF$^+$) are equally strong in terms of their LP-relaxations, i.e., for all instances of the DCSTP, we have:*

$$v_{\text{LP}}(CF^+) = v_{\text{LP}}(UF^+).$$

Proof. By using the same linear transformation as in the proof of Proposition 2, we obtain

$$\sum_{(i,j)\in A} u_{ij}^{qt} - \sum_{(j,i)\in A} u_{ji}^{qt} = \tag{11}$$

$$\left(\sum_{(i,j)\in A} f_{ij}^{q} - \sum_{(j,i)\in A} f_{ji}^{q}\right) + \left(\sum_{(i,j)\in A} f_{ij}^{t} - \sum_{(j,i)\in A} f_{ji}^{t}\right) - 2\left(\sum_{(i,j)\in A} m_{ij}^{qt} - \sum_{(j,i)\in A} m_{ji}^{qt}\right), \tag{12}$$

for any $i \in V,\ q \in T \setminus \{0\},\ t \in T \setminus \{0, q\},\ q < t$. Let us now assume that (8) holds. We distinguish four cases:

1. $i \in V \setminus \{0, t, q\}$:

$$0 \geq 2\left(\sum_{(i,j)\in A} m_{ij}^{qt} - \sum_{(j,i)\in A} m_{ji}^{qt}\right) = 0 + 0 - \left(\sum_{(i,j)\in A} u_{ij}^{qt} - \sum_{(j,i)\in A} u_{ji}^{qt}\right).$$

2. $i = 0$:

$$2 \geq 2\left(\sum_{(i,j)\in A} m_{ij}^{qt} - \sum_{(j,i)\in A} m_{ji}^{qt}\right) = 1 + 1 - \left(\sum_{(i,j)\in A} u_{ij}^{qt} - \sum_{(j,i)\in A} u_{ji}^{qt}\right).$$

3. $i = t$:

$$0 \geq 2\left(\sum_{(i,j)\in A} m_{ij}^{qt} - \sum_{(j,i)\in A} m_{ji}^{qt}\right) = -1 + 0 - \left(\sum_{(i,j)\in A} u_{ij}^{qt} - \sum_{(j,i)\in A} u_{ji}^{qt}\right).$$

4. $i = q$:

$$0 \geq 2\left(\sum_{(i,j)\in A} m_{ij}^{qt} - \sum_{(j,i)\in A} m_{ji}^{qt}\right) = 0 - 1 - \left(\sum_{(i,j)\in A} u_{ij}^{qt} - \sum_{(j,i)\in A} u_{ji}^{qt}\right).$$

Hence, inequalities (10) are satisfied.
Similarly, using the same relationship given by (11), and by distinguishing the same four cases, one can show that inequalities (10) can be transformed into (8). □

With Propositions 2 and 3 we have related the LP relaxations of the common and uncommon flow models. We present next an equality that relates the variables u of the uncommon flow model with the variables g of the (SF) model.

$$g_{ij}^{qt} + g_{ji}^{qt} = u_{ij}^{qt} + u_{ji}^{qt} \qquad q, t \in T \setminus \{0\}, q < t, \{i, j\} \in E \tag{13}$$

Equality (13) follows from the fact that an edge is used in one of the directions by the g flow model if and only if it is used by the uncommon flow u.

Consider the inequality (9e) for a given 4-tuple (i, j, q, t). If we add the same constraint for the tuple (j, i, q, t) we obtain:

$$f_{ij}^{q} + f_{ij}^{t} + f_{ji}^{q} + f_{ji}^{t} + u_{ij}^{qt} + u_{ji}^{qt} \leq 2(a_{ij} + a_{ji}), \quad q, t \in T \setminus \{0\}, q < t, \{i, j\} \in E \tag{14}$$

By using (13) we obtain (6). Hence, while constraints (6) strengthen the (SF) model, they are implied by the common/uncommon flow formulation.

Two observations are in order. First, we note that (6) is obtained after adding two constraints, (9e), from the uncommon flow model. Thus, before obtaining (6), we have weakened the LP relaxation of the uncommon flow model. Secondly, in the same way we have derived (6), we can derive other inequalities in the space of the f and g variables by combining other inequalities from the uncommon flow model. We leave the investigation of these inequalities for future research.

4 Computational Results

In this section, we discuss the results obtained in our computational study. All formulations have been implemented using julia 1.8.0 and JuMP 1.0 (Lubin et al., 2023) in combination with CPLEX 22.1 for solving MILPs. All experiments have been performed on a single core of a standard notebook with an i7-10510U CPU with 1.80GHz and 16GB RAM. We used

benchmark instances with $|V| \in \{20, 30\}$ that are frequently used for testing approaches for hop- and distance-constrained tree problems (see, e.g., Gouveia et al. (2011, 2015)) in which nodes are randomly placed in a grid. We use the first instance from each set, define the first $|T|$ nodes ($|T| \in \{5, 10\}$) as terminals and use the first terminal as root node 0 in our formulations. We consider instances with random costs (TR) and Euclidean costs (TC, TE) and diameter values $D \in \{3, 4, 5\}$. The main difference between the latter two is the location of the first terminal node (in the center for TC and at the corner for TE).

Note that this study does not aim to improve the computational state-of-the-art for solving diameter-constrained Steiner trees which would likely by obtained by (straightforward) adaptations of the approach from Gouveia et al. (2011) originally proposed for spanning trees to the Steiner tree variant. Instead, we want to understand properties of and the differences between the formulations discussed in the previous sections and shed some light on the question whether using them within dedicated Benders decomposition approaches (that avoid explicitly considering the large number of flow variables) may be promising. Thus, we mainly focus on analyzing the lower bounds obtained from their LP-relaxations. In the second part of our study, we also discuss results obtained from solving the MILPs as well as on applying the automatic or annotated Benders decomposition method of CPLEX to them.

4.1 Comparison of Lower Bounds

Tables 1 and 2 provide LP-relaxation bounds and runtimes needed to solve the LP-relaxations, respectively. We consider the following formulations: (SF), (SF)+(4), (SF)+(4)-(5), (SF)+(4)-(6), and also (CF) and (CF$^+$). Given the results of Propositions 2 and 3, we can deduce the bounds for the models (UF) and (UF$^+$), respectively, which is why we do not explicitly report them. Each row in Tables 1 and 2 corresponds to a single instance. The first four columns in these tables indicate: the benchmark set (TR, TC, or TE), the number of nodes, the diameter D and the number of terminals $|T|$ in each instance, respectively. The value of the optimal solution is shown in column v_{IP} of Table 1, followed by the LP-relaxation bounds of each of the six models mentioned above. Bold values in Table 1 indicate the best obtained bounds (rounded to two decimal places). The times reported in Table 2 are runtimes (in seconds rounded to one decimal place) needed to solve the respective formulations after relaxing the binary variables.

The obtained results indicate that each of the constraints introduced in Section 2.1 helps in improving the quality of lower bounds of the standard (SF) model. We also observe that even the strongest model based on pairwise diameter constraints, namely (SF)+(4)-(6) can provide bounds which are weaker than the ones obtained by the (CF) model (see, e.g., instances TR-30-*-10). On the other hand, we do not have examples in which the bounds of the (CF) model are weaker than those of the (SF)+(4)-(6). Hence, it remains an open question whether the LP-relaxation bounds of the (CF) model theoretically dominate the LP-relaxation bounds of the (SF)-based models.

When focusing on the relative improvements of the lower bounds with respect to the basic reference value, which is $v_{\text{LP}}(SF)$, we notice that these improvements are not very significant in most cases. While the two largest improvements are 13.8% (TR, $|V| = 30$, $|T| = 10$, $D = 3$) and 8.2% (TR, $|V| = 20$, $|T| = 10$, $D = 3$) they are below 3.5% in all other cases. Concerning the runtimes, we have several rather surprising results: the fastest model is the basic (SF) formulation, and adding constraints (4) may increase the runtimes by an order of magnitude. Addition of constraints (5) (or (5)-(6)) may further increase the runtimes by another order of magnitude. Given the relatively small improvement in the quality of LP-relaxation bounds achieved by adding these constraints, we conclude that there is a little value of adding these constraints within a branch-and-bound procedure. Indeed, it is more likely that MIP solvers will

much faster improve the lower bounds by starting from the basic model (SF) while leveraging on branching strategies. When it comes to (CF) and (CF$^+$) formulations, we notice that quite often the runtimes are comparable to those of the (SF) model, but they can also be of an order of magnitude higher. Hence, the conclusion is the same: if we were to use the proposed models and solve them with an off-the-shelf solver, the basic model (SF) will be most likely the best performing one, as the runtime would be in favor of the (SF) model when resolving the trade-off between the quality of lower bounds and the size of the branching tree.

4.2 Branch-and-Benders Cut Implementations

We now turn our attention to alternative solution approaches. We namely consider branch-and-Benders-cut implementations for the proposed formulations and compare them with the traditional branch-and-bound approaches in which the LP-relaxations of the flow-based models are solved at every node of the branching tree. For both types of implementations, we use Cplex as off-the-shelf solver. Cplex parameter `Benders strategy` has to be set to 3, in order to activate the `automatic Benders` decomposition (i.e., the branch-and-Benders implementation), in which all flow variables are projected out from the respective model, and Benders feasibility and optimality cuts are generated in terms of a variables. We point out that in Cplex's implementation of branch-and-Benders-cut, all continuous variables are projected out and considered as part of the Benders subproblem. If the constraint matrix admits a block angular structure, the Benders LP subproblem is separated into multiple, smaller LPs, each of them is solved independently, and multiple Benders cuts are generated in each iteration (one per each subproblem). If, on the other hand, the subproblem is not separable, a single Benders cut is generated per each iteration. Thus, one can resort to `Benders strategy` equal to 2 (also known as `annotated Benders`) in order to explicitly keep certain continuous variables in the master problem (which may help Cplex to identify multiple separable subproblems).

In the Benders subproblem derived from the formulation (SF), the flow f is separable by each terminal. In addition, the flow g is separable by each pair (q, t) of terminals, $q, t \in T \setminus \{0\}$, $q < t$, and hence the underlying LPs can be solved very fast. This separability property allows to add multiple violated Benders cuts in the same iteration and also reduces the number of iterations needed to converge to the LP-optimal solution. The remaining models unfortunately do not admit this property. The separability between f and g is destroyed after adding constraints (4) to (SF), since f and g flows can no longer be solved independently. Similar arguments apply to models (CF$^+$) and (UF$^+$) where all the model constraints link m, respectively u, variables with f, and hence, the separability property does not hold. This leads to a very poor performance of all models (except (SF)) when solved using the automatic Benders strategy. To circumvent this problem of non-separability, we resort to the annotated Benders strategy. In particular, we ensure that flow variables f stay in the master problem. As a consequence, the subproblems decompose per terminal pair $q, t \in T \setminus \{0\}$, $q < t$ for formulations (SF)+(4), (CF), (CF$^+$), (UF), and (UF$^+$). This modification comes at the cost of keeping additional $|T||A|$ variables in the master problem. On the contrary, the Benders subproblem remains non-separable for formulation (SF)+(4)–(6) due to the bidirectional forcing constraints (5). As we will see below, this separability property plays an important role when evaluating the efficiency of the proposed formulations, when it comes to branch-and-Benders-cut implementations.

Table 3 provides summary of obtained results, in which each row corresponds to a single instance. Both, branch-and-bound and branch-and-Benders-cut are given a timelimit of 1h. The table reports runtimes shown in seconds (if the instance is solved to optimality within the timelimit), or gaps at termination (shown in parenthesis). The gaps are computed relative to the best-known primal bound. A gap of 100% indicates that no feasible solution has been identified

Table 1: LP-bounds. Bold values indicate the best obtained bounds per each instance.

					$v_{\mathrm{LP}}((SF)+\cdot)$				$v_{\mathrm{LP}}(\cdot)$	
Set	$\lvert V\rvert$	D	$\lvert T\rvert$	v_{IP}	-	(4)	(4)-(5)	(4)-(6)	(CF)	(CF$^+$)
R	20	3	5	84	74.95	75.86	**76.00**	**76.00**	**76.00**	**76.00**
R	20	4	5	63	61.33	61.33	**61.67**	**61.67**	**61.67**	**61.67**
R	20	5	5	63	**55.00**	**55.00**	**55.00**	**55.00**	**55.00**	**55.00**
R	20	3	10	210	147.72	154.11	159.20	159.40	159.66	**159.78**
R	20	4	10	110	**110.00**	**110.00**	**110.00**	**110.00**	**110.00**	**110.00**
R	20	5	10	110	105.67	**106.00**	**106.00**	**106.00**	**106.00**	**106.00**
R	30	3	5	83	65.30	66.15	66.22	**66.75**	**66.75**	**66.75**
R	30	4	5	48	45.50	45.50	45.50	45.75	45.75	**46.00**
R	30	5	5	36	**36.00**	**36.00**	**36.00**	**36.00**	**36.00**	**36.00**
R	30	3	10	217	141.78	148.58	156.44	160.84	161.29	**161.40**
R	30	4	10	121	103.91	104.73	105.90	107.92	108.02	**108.21**
R	30	5	10	104	87.16	87.34	87.50	88.27	88.28	**88.30**
C	20	3	5	167	**167.00**	**167.00**	**167.00**	**167.00**	**167.00**	**167.00**
C	20	4	5	167	**167.00**	**167.00**	**167.00**	**167.00**	**167.00**	**167.00**
C	20	5	5	167	**167.00**	**167.00**	**167.00**	**167.00**	**167.00**	**167.00**
C	20	3	10	298	252.09	255.40	259.50	**261.00**	**261.00**	**261.00**
C	20	4	10	235	227.78	229.56	232.50	232.50	232.50	**233.13**
C	20	5	10	225	218.83	220.55	**221.59**	**221.59**	**221.59**	**221.59**
C	30	3	5	106	**104.00**	**104.00**	**104.00**	**104.00**	**104.00**	**104.00**
C	30	4	5	104	**104.00**	**104.00**	**104.00**	**104.00**	**104.00**	**104.00**
C	30	5	5	104	**104.00**	**104.00**	**104.00**	**104.00**	**104.00**	**104.00**
C	30	3	10	294	277.10	278.30	**279.25**	**279.25**	**279.25**	**279.25**
C	30	4	10	252	**252.00**	**252.00**	**252.00**	**252.00**	**252.00**	**252.00**
C	30	5	10	246	245.00	245.25	**245.33**	**245.33**	**245.33**	**245.33**
E	20	3	5	144	132.44	132.44	132.44	**132.50**	**132.50**	**132.50**
E	20	4	5	128	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**
E	20	5	5	128	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**
E	20	3	10	279	247.88	**249.00**	**249.00**	**249.00**	**249.00**	**249.00**
E	20	4	10	221	**221.00**	**221.00**	**221.00**	**221.00**	**221.00**	**221.00**
E	20	5	10	213	**213.00**	**213.00**	**213.00**	**213.00**	**213.00**	**213.00**
E	30	3	5	140	132.44	132.44	132.44	**132.50**	**132.50**	**132.50**
E	30	4	5	128	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**
E	30	5	5	128	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**	**128.00**
E	30	3	10	279	247.66	**249.00**	**249.00**	**249.00**	**249.00**	**249.00**
E	30	4	10	221	**221.00**	**221.00**	**221.00**	**221.00**	**221.00**	**221.00**
E	30	5	10	221	**213.00**	**213.00**	**213.00**	**213.00**	**213.00**	**213.00**

Table 2: Times (in seconds) for solving LP relaxations.

				$t_{\mathrm{LP}}((SF)+\cdot)$				$t_{\mathrm{LP}}(\cdot)$	
Set	$\|V\|$	D	$\|T\|$	-	(4)	(4)-(5)	(4)-(6)	(CF)	(CF$^+$)
C	20	3	5	0.2	0.6	0.7	0.7	0.2	0.6
C	20	4	5	0.2	0.3	0.8	0.6	0.3	0.2
C	20	5	5	0.1	0.4	0.6	0.6	0.2	0.3
C	20	3	10	3.2	35.1	216.6	285.0	29.7	59.3
C	20	4	10	1.3	32.9	131.5	100.2	16.6	25.5
C	20	5	10	0.8	18.5	60.7	43.6	6.0	11.8
C	30	3	5	0.4	1.2	3.9	2.6	2.0	2.9
C	30	4	5	0.4	3.6	5.0	4.0	0.8	1.2
C	30	5	5	0.3	1.4	3.9	2.9	0.7	1.3
C	30	3	10	16.3	107.3	1818.6	1381.6	141.8	222.4
C	30	4	10	23.0	110.2	2649.6	1134.8	70.8	160.7
C	30	5	10	17.7	63.4	2502.0	1241.9	35.0	49.7
E	20	3	5	0.1	0.3	0.6	0.8	0.8	0.7
E	20	4	5	0.1	0.6	1.0	1.0	0.1	0.1
E	20	5	5	0.1	0.4	0.7	0.3	0.1	0.2
E	20	3	10	3.7	49.3	221.0	507.7	21.5	38.4
E	20	4	10	3.3	65.2	142.0	147.5	8.3	15.8
E	20	5	10	3.2	131.2	258.0	183.1	5.2	5.7
E	30	3	5	1.9	13.9	16.3	24.4	2.0	7.4
E	30	4	5	0.4	2.0	4.0	5.6	1.5	2.2
E	30	5	5	0.3	1.5	3.4	3.4	1.0	1.9
E	30	3	10	58.5	571.7	2159.3	1057.8	187.1	525.3
E	30	4	10	15.5	92.8	804.6	720.1	54.5	75.6
E	30	5	10	11.1	80.6	966.4	751.8	36.9	51.8
R	20	3	5	0.2	0.3	0.6	0.8	1.1	1.6
R	20	4	5	0.1	0.5	0.6	0.6	0.4	1.0
R	20	5	5	0.1	0.2	0.2	0.5	0.1	0.1
R	20	3	10	2.1	24.3	211.2	396.0	49.4	126.1
R	20	4	10	0.9	10.0	14.0	29.4	5.7	18.9
R	20	5	10	0.8	13.5	34.8	30.7	7.8	10.2
R	30	3	5	1.0	2.6	5.7	3.6	4.8	5.5
R	30	4	5	0.8	2.4	3.9	5.4	3.2	5.0
R	30	5	5	0.1	0.5	1.3	0.5	0.5	0.8
R	30	3	10	19.4	740.3	1683.0	1979.4	816.4	1696.6
R	30	4	10	13.8	97.4	1778.1	2243.6	302.6	491.8
R	30	5	10	12.2	66.8	2180.3	1881.9	101.3	297.7

within the timelimit. We denote by $t_{\text{IP}}(\cdot)$ the runtime of the branch-and-bound implementation, and by $t_{\text{Ben}}(\cdot)$ the runtime of the branch-and-Benders-cut implementation. We compare the following formulations: the standard flow-based model (SF), and its two extensions, namely (SF)+(4) and (SF)+(4)-(6). In addition, we consider (CF$^+$) and (UF$^+$) formulations. We do not report results for using Benders decomposition on model (SF)+(4)-(6) as this approach failed to solve almost all instances (and resulted in very high optimality gaps) due to the non-decomposable subproblem leading to at most one Benders cut per iteration.

When comparing the solution times t_{IP} of five compact models, we observe a direct correlation between the LP solution times and the overall performance, i.e., the stronger LP-relaxation bounds do not pay off when the models are solved in their compact form. In particular, adding strengthening constraints (4)-(6) to the standard model (SF) does not help to reduce the overall runtimes. On the contrary, already by inserting only constraints (4) to (SF), for some of the instances we observe an order of magnitude increase in runtime. While only a single instance out of 36 instances remains unsolved within one hour when (SF) model is used, this number increases to five, respectively 11, for the model (SF)+(4) and (SF)+(4)-(6), respectively. The models (CF$^+$) and (UF$^+$) perform similarly, and six, respectively, five instances remain unsolved within 1h.

Comparing the branch-and-Benders-cut implementations versus their "IP-counterparts", we observe significant speed-ups, especially for non-trivial instances. So, for example, using Benders cuts, all instances from our benchmark set can be solved to optimality by the model (SF) and the model (SF)+(4) solves all but one instance. For the remaining two models, (CF$^+$) and (UF$^+$), three instances remain unsolved. We observe that, when Benders cuts are used, the winner in terms of performance is the model (SF). This result is quite surprising, and we believe that several major factors explain such a performance: first, the model (SF) already provides strong lower bounds and the quality of these bounds is preserved when Benders cuts are dynamically separated. When comparing t_{IP} times, we have already seen that the minor relative improvement in the quality of lower bounds achieved by adding strengthening inequalities (or considering alternative formulations) does not pay off in the overall branch-and-bound context. Finally, the major reason why the Benders cuts do not make the models (SF)+(4), (SF)+(4)-(6), (CF$^+$) and (UF$^+$) competitive against (SF) lies in the fact that the size of the master problem is enlarged by a factor of $|T|$, both in terms of decision variables and constraints. Indeed, to maintain the separability property of the Benders subproblem, flow variables f are kept in the master problem, which significantly slows down every single LP-iteration of the master problem. On the other hand, leaving the variables f in the subproblem destroys the separability and makes the overall performance even worse (due to the excessive times needed to solve a single LP of the Benders subproblem, and slow convergence due to generation of a single cut per iteration).

To summarize, we conclude that Benders cuts significantly improve the performance of compact models, while preserving the quality of lower bounds. However, there is still a significant difference in the performance of branch-and-Benders-cut implementations, which does not only depend on the strength of the underlying model, but also on the separability of the respective Benders subproblems. If this property is not preserved, or is just partially preserved, the overall performance can be negatively affected.

5 Conclusions

In this work we have studied several flow-based formulations for modeling the spanning, respectively, Steiner tree problem whose diameter does not exceed a given parameter D. We have started from a problem relaxation in which we model a Steiner arborescence rooted at a randomly chosen terminal node and ensure that the length of each path between the root and any

Table 3: IP solution times (values in parenthesis are gaps in percent after 1h in case not solved).

				$t_{\text{IP}}((SF)+\cdot)$			$t_{\text{IP}}(\cdot)$		$t_{\text{Ben}}(\cdot)$			
Set	$\|V\|$	D	$\|T\|$	-	(4)	(4)-(6)	(CF^+)	(UF^+)	(SF)	(SF)+(4)	(CF^+)	(UF^+)
C	20	3	5	0.2	0.5	0.4	0.4	0.6	0.3	0.2	0.2	0.3
C	20	4	5	0.1	0.7	0.4	0.5	0.3	0.2	0.1	0.2	0.2
C	20	5	5	0.2	0.3	0.4	0.2	0.3	0.2	0.1	0.2	0.2
C	20	3	10	2765.9	(8.7)	(100.0)	(11.1)	(15.4)	33.7	712.1	(9.1)	(8.2)
C	20	4	10	18.1	628.4	(100.0)	71.1	73	2.2	6.0	42	20.1
C	20	5	10	3.2	195.1	(100.0)	24.7	78.5	2.3	4.9	19.5	13.2
C	30	3	5	0.9	1.7	4.4	2.8	3.9	0.4	0.5	1.8	1.1
C	30	4	5	0.3	0.3	3.9	0.7	1.1	0.3	0.5	0.4	0.5
C	30	5	5	0.2	0.2	4.4	0.6	1	0.3	0.4	0.4	0.5
C	30	3	10	144.1	2794.6	(22.6)	(21.4)	1831.5	7.1	63.1	2087.3	1813.8
C	30	4	10	8.6	76.7	440.2	102.6	214.2	2.4	6.1	28.8	81.1
C	30	5	10	10.1	160.4	263	24.1	200.5	2.2	9.0	10.3	10.2
E	20	3	5	0.4	4.1	2.4	3	1.8	0.3	0.4	1.9	1.2
E	20	4	5	0.1	0.3	0.5	0.2	0.2	0.2	0.1	0.2	0.1
E	20	5	5	0.1	0.1	0.5	0.3	0.2	0.1	0.2	0.1	0.1
E	20	3	10	102.4	1686.8	(10.7)	948.4	1240.4	5.1	62.3	417.1	227.1
E	20	4	10	0.6	2.8	12.6	5.6	19.5	0.9	1.9	2.5	2.8
E	20	5	10	0.5	3.3	12	3.4	19.7	0.9	1.3	2.1	2.6
E	30	3	5	1.1	7.8	16.2	20.2	14.5	0.7	1.2	7.5	3.4
E	30	4	5	1.9	0.4	10.3	3.2	5.4	0.6	0.3	0.5	0.4
E	30	5	5	0.4	0.3	2.5	0.7	1.3	0.5	0.4	0.3	0.4
E	30	3	10	521.5	(5.1)	(100.0)	(8.1)	(7.1)	18.9	231.1	(5.1)	(7.9)
E	30	4	10	25.6	21.5	1697.9	122.9	395.4	1.6	6.2	16.5	37.9
E	30	5	10	5.0	34.9	382.4	63.8	466.6	1.9	6.8	13.3	22.4
R	20	3	5	0.5	3.2	2.9	3.7	3	0.3	0.3	2.3	2.5
R	20	4	5	0.1	0.6	1.6	1.6	0.6	0.1	0.1	0.6	0.5
R	20	5	5	0.4	3.5	3	1.7	3.6	0.2	0.3	0.3	0.3
R	20	3	10	2583.9	(27.4)	(67.0)	2979.4	2765.2	77.9	254.1	976.7	679.0
R	20	4	10	0.8	5.3	33.3	6.7	27.5	0.8	1.6	2.0	1.8
R	20	5	10	1.4	18.2	(2.7)	156.4	92	1	1.9	5.8	8.4
R	30	3	5	5.5	25.4	52.9	73.3	33.5	1.9	3.4	13.5	5.2
R	30	4	5	1.1	2.4	9.2	15.2	13.6	0.4	0.8	3.4	1.1
R	30	5	5	0.2	0.3	0.8	0.7	0.9	0.2	0.2	0.3	0.3
R	30	3	10	(37.9)	(44.0)	(100.0)	(100.0)	(38.5)	1571.6	(10.9)	(65.2)	(28.0)
R	30	4	10	2139.4	2586.5	(100.0)	(46.2)	(43.5)	12.7	133.5	2058.5	645.6
R	30	5	10	405.2	(9.7)	(100.0)	(21.7)	(18.4)	20.2	181.2	334.9	662.4

other terminal does not exceed D. We have considered three ways to extend this formulation in order to obtain a valid model. The first approach is a standard formulation, (SF), in which additional flow variables are introduced to model pairwise connections between the terminals. The model (SF) is then enhanced by several families of valid inequalities. The second approach is based on the concept of common flows which has been introduced to strengthen LP-relaxations for flow-based formulations of Steiner trees (Polzin and Daneshmand, 2001). We show that common flows can be used not only to strengthen the lower bounds, but also to enforce diameter constraints between pairs of terminals. Finally, we have introduced an alternative idea of *uncommon flows*, and shown that also the uncommon flows can be used to limit the pairwise distance between pairs of terminals.

We have compared the proposed models both theoretically and empirically. While the valid inequalities added to (SF) improve the quality of LP-relaxation bounds, and the models derived from common/uncommon flows in principle provide even stronger bounds, the runtimes needed to solve their LP-relaxations are prohibitive and do not make them applicable in practical context. In the second half of our empirical study we have examined the computational performance of the proposed formulations when they are embedded in a branch-and-Benders cut framework. We have seen that projecting out flow variables significantly improves the performance of compact models, while preserving the quality of lower bounds. However, we have observed a significant difference in the performance of branch-and-Benders cuts, as they are affected by the separability of the underlying Benders subproblems. When the separability does not or only partially holds, the overall performance is negatively affected. Hence, the best performing approach is derived from the model (SF) in combination with a branch-and-Benders cut implementation.

The results of our study can trigger further research on strong flow-based formulations for diameter constrained spanning/Steiner trees and related network design problems. Instead of solving LPs to derive Benders cuts, it would be interesting to investigate combinatorial procedures (in particular for the separation of integer infeasible points). In addition, departing from common/uncommon flows and along the lines of disaggregation techniques proposed in Croxton et al. (2007); Gendron and Gouveia (2017), we may obtain new models with even better LP-relaxation bounds. Disaggregations can be based on using position-indexed arc-variables as in Gouveia (1998); Gendron and Gouveia (2017), or on the choice of the root node as in Gouveia et al. (2006).

References

N. R. Achuthan, Lou Caccetta, P. Caccetta, and James F. Geelen. Computational methods for the diameter restricted minimum weight spanning tree problem. *Australasian Journal of Combinatorics*, 10:51–72, 1994.

A. Balakrishnan, T. L. Magnanti, and R. T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740, 1989.

Huynh Thi Thanh Binh, Robert I. McKay, Nguyen Xuan Hoai, and Nguyen Duc Nghia. New heuristic and hybrid genetic algorithm for solving the bounded diameter minimum spanning tree problem. In Franz Rothlauf, editor, *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009*, pages 373–380. ACM, 2009.

Abraham Bookstein and Shmuel T. Klein. Compression of correlated bit-vectors. *Information Systems*, 16(4):387–400, 1991.

Mervat Chouman, Teodor Gabriel Crainic, and Bernard Gendron. Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science*, 51(2):650–667, 2017.

Claudio Contardo, Jean-François Cordeau, and Bernard Gendron. A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optimization*, 10 (4):263–295, 2013.

Keely L. Croxton, Bernard Gendron, and Thomas L. Magnanti. Variable disaggregation in network flow problems with piecewise linear costs. *Operations Research*, 55(1):146–157, 2007.

Narsingh Deo and Ayman Abdalla. Computing a diameter-constrained minimum spanning tree in parallel. In Giancarlo Bongiovanni, Rossella Petreschi, and Giorgio Gambosi, editors, *Algorithms and Complexity*, pages 17–31, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

Andréa C. dos Santos, Abilio Lucena, and Celso C. Ribeiro. Solving diameter constrained minimum spanning tree problems in dense graphs. In Celso C. Ribeiro and Simone L. Martins, editors, *Experimental and Efficient Algorithms, Third International Workshop, WEA 2004, Angra dos Reis, Brazil, May 25-28, 2004, Proceedings*, volume 3059 of *Lecture Notes in Computer Science*, pages 458–467. Springer, 2004.

Antonio Frangioni and Bernard Gendron. 0-1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, 157(6):1229–1241, 2009.

M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. ISBN 0-7167-1044-7.

Bernard Gendron and Luis Eduardo Neves Gouveia. Reformulations by discretization for piecewise linear integer multicommodity network flow problems. *Transportation Science*, 51(2): 629–649, 2017.

Bernard Gendron and Frédéric Semet. Formulations and relaxations for a multi-echelon capacitated location–distribution problem. *Computers & Operations Research*, 36(5):1335–1355, 2009.

Luis Gouveia and Thomas L. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks*, 41(3):159–173, 2003.

Luis Gouveia, Thomas L. Magnanti, and Cristina Requejo. A 2-path approach for odd-diameter-constrained minimum spanning and Steiner trees. *Networks*, 44(4):254–265, 2004.

Luis Gouveia, Luidi Simonetti, and Eduardo Uchoa. Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming*, 128(1-2):123–148, 2011.

Luis Gouveia, Markus Leitner, and Ivana Ljubić. The two-level diameter constrained spanning tree problem. *Mathematical Programming*, 150(1):49–78, 2015.

Luis Gouveia, Markus Leitner, and Ivana Ljubić. A polyhedral study of the diameter constrained minimum spanning tree problem. *Discrete Applied Mathematics*, 285:364–379, 2020.

Luis Eduardo Neves Gouveia. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. *INFORMS Journal on Computing*, 10 (2):180–188, 1998.

Luis Eduardo Neves Gouveia, Thomas L. Magnanti, and Cristina Requejo. An intersecting tree model for odd-diameter-constrained minimum spanning and Steiner trees. *Annals of Operations Research*, 146(1):19–39, 2006.

Martin Gruber, Jano I. van Hemert, and Günther R. Raidl. Neighbourhood searches for the bounded diameter minimum spanning tree problem embedded in a VNS, EA, and ACO. In Mike Cattolico, editor, *Genetic and Evolutionary Computation Conference, GECCO 2006, Proceedings, Seattle, Washington, USA, July 8-12, 2006*, pages 1187–1194. ACM, 2006.

Ivana Ljubić. Solving Steiner trees: Recent advances, challenges, and perspectives. *Networks*, 77(2):177–204, 2021.

Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. Jump 1.0: Recent improvements to a modeling language for mathematical optimization, 2023.

Abilio Lucena, Celso C. Ribeiro, and Andréa C. Santos. A hybrid heuristic for the diameter constrained minimum spanning tree problem. *Journal of Global Optimization*, 46(3):363–381, 2010.

Thiago F. Noronha, Celso C. Ribeiro, and Andréa C. Santos. Solving diameter-constrained minimum spanning tree problems by constraint programming. *International Transactions in Operational Research*, 17(5):653–665, 2010.

T. Polzin and S. V. Daneshmand. A comparison of Steiner tree relaxations. *Discrete Applied Mathematics*, 112(1-3):241–261, 2001.

Günther R Raidl and Bryant A Julstrom. Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem. In *Proceedings of the 2003 ACM symposium on applied computing*, pages 747–752, 2003.

C. Requejo and E Santos. Greedy heuristics for the diameter-constrained minimum spanning tree problem. *Journal of Mathematical Sciences*, 161:930–943, 2009.

Andréa Cynthia Santos, Christophe Duhamel, and Rafael Andrade. *Trees and Forests*, pages 1–27. Springer International Publishing, Cham, 2016.

Alok Singh and Ashok K Gupta. Improved heuristics for the bounded-diameter minimum spanning tree problem. *Soft Computing*, 11:911–921, 2007.

Wolfgang Steitz. New heuristic approaches for the bounded-diameter minimum spanning tree problem. *INFORMS Journal on Computing*, 27(1):151–163, 2015.

Babacar Thiongane, Jean-François Cordeau, and Bernard Gendron. Formulations for the nonbifurcated hop-constrained multicommodity capacitated fixed-charge network design problem. *Computers & Operations Research*, 53:1–8, 2015.

K.-H. Vik, P. Halvorsen, and C. Griwodz. Multicast tree diameter for dynamic distributed interactive applications. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 1597–1605, 2008. doi: 10.1109/INFOCOM.2008.220.