Noname manuscript No.
(will be inserted by the editor)

Integer Programming Formulations for the Shared Multicast Tree Problem

Marika Ivanova · Dag Haugland

Received: date / Accepted: date

Abstract We study the Shared Multicast Tree (SMT) problem in wireless networks. To support a multicast session between a set of network nodes, SMT aims to establish a wireless connection between them, such that the total energy consumption is minimized. All destinations of the multicast message must be connected, while non-destinations are optional nodes that can be used to relay messages. The objective function reflecting power consumption distinguishes SMT clearly from the traditional minimum Steiner tree problem.

We develop two integer programming formulations for SMT. Both models are subsequently extended and strengthened. Theorems on relations between the LP bounds corresponding to the models are stated and proved. As the number of variables in the strongest formulations is a polynomial of degree four in the number of network nodes, the models are impractical for computing lower bounds in instances beyond a fairly small size, and therefore a constraint generation scheme is developed. Results from computational experiments with the models demonstrate good promise of the approaches taken.

Keywords Wireless communication, multicast tree, Steiner tree, LP bound, valid inequalities

1 Introduction

Wireless ad hoc networks (WANETs) are suitable in various practical applications in both civil and military sector, where a communication infrastructure

M. Ivanova
Department of Informatics, University of Bergen, Norway
Tel.: +47 94277070
E-mail: Marika.Ivanova@uib.no

D. Haugland
Department of Informatics, University of Bergen, Norway
Tel.: +47 55584033
E-mail: Dag.Haugland@uib.no

is either absent or cannot be relied on. Owing to their quick deployment and simple configuration, WANETs are useful in emergency operations such as natural disaster relief and military command and control. Recently, the WANET concept has been introduced in ad hoc smart lighting in homes and in offices, as well as ad hoc street light networks, where the control includes adjusting dimmable lights.

The purpose of a multicast communication session in a WANET is to route information from a sending device to a set of receiving devices. Connection links are established by assigning sufficient power to the sending devices. Since the devices typically use batteries as power supply, they are heavily energy-constrained. Therefore, power efficiency is an important measure in designing WANETs. Individual devices work as transceivers, which means that they have the ability to both transmit and receive a signal. Moreover, the power level of a device can be dynamically adjusted during a multicast session. Montemanni and Leggieri (2011) name several applications of wireless sensor networks where power saving is essential.

Unlike wired networks, where signal passing takes place along predefined links, nodes in WANETs use omnidirectional antennas, and hence a message reaches all nodes within the communication range of its sender. This range is determined by the power assigned to the sender, which is the maximum rather than the sum of the powers necessary to reach all intended receivers. This feature is often referred to as the *wireless advantage* (Wieselthier et al. 2000).

1.1 Related problems

A well-known and extensively studied problem in the network design literature is the Minimum Energy Broadcast (MEB) problem. Given a set of wireless devices with one designated source node, the goal is to assign power to individual nodes. The power assignments determine the communication ranges, and induce a broadcast tree such that a signal initiated by the source reaches, directly or via other nodes, all the remaining nodes. In the MEB problem, the task is to find power assignments enabling such a signal broadcast while minimizing the total energy consumption. A natural and frequently addressed extension of MEB, is the Minimum Energy Multicast (MEM) problem. Analogous to the difference between the Minimum Spanning Tree and the Minimum Steiner Tree problems, MEM is distinguished from MEB in that not all nodes in the network are destinations of the multicast message. Hence, the more general problem amounts to finding a Steiner tree spanning the destination nodes such that the sum of the power assignments is minimized. Non-destination nodes are allowed in the Steiner tree, and will be included to the extent that they contribute to reduced power consumption.

In certain practical settings, MEB fails to reflect reality adequately. Consider an application where more than one node can act as a source. The optimal broadcast trees corresponding to two different sources are not necessarily identical, and the optimal broadcast trees must be calculated separately for every possible source node. In order to relay a received signal correctly, the nodes must be able to recognize which node initiated the received signal, and therefore also determine which broadcast tree to use. Further, which power level to be set must be assessed by the relaying node. Such overhead calculations require additional energy, and enforcing the communication devices to accomplish them may even be impractical.

The idea behind the Minimum Shared Broadcast Tree (SBT) problem is to maintain a single, shared broadcast tree, to be applied regardless of which node is the signal source. Such a tree is not necessarily optimal for any of the individual sources, but routing at each node is considerably simplified. Provided that a shared broadcast tree is used, the nodes are no longer required to identify the source of the message in order to set a correct power level. Instead, only the immediate neighbour from which the signal was received must be recognized. The objective function in SBT captures not only the power levels of the nodes, but depends also on how frequently a node actually transmits with a certain power level.

The SBT and SMT problems resemble the more frequently studied Range Assignment Problem (RAP) in the sense that they all ask for an undirected source-independent tree of minimum power. However, RAP concerns cases where a connected topology must be established by means of *bidirectional* links, with the feature that power must be assigned to both the sending and the receiving device in each link. This scenario induces a spanning tree problem, where costs are incurred only for edges that are the most expensive ones incident to any of their end nodes.

1.2 Contributions

Contributions from the present work include two sets of Integer Programming (IP) models for SMT, presented in order of increasing lower bounding capabilities. By rigorous analysis of the models, we prove relations between the bounds offered by the corresponding continuous relaxations. Intended for instances beyond the size that can be solved to optimality, we also develop a constraint generation procedure by which strong bounds can be computed.

When proven optimality is out of reach, effective methods for computing near-optimal solutions are crucial. A heuristic method tailored for SMT is another contribution from the current text, which thus demonstrates how recent progress on the Minimum Steiner Tree problem can be transferred to the problem under study.

1.3 Scope and structure of the paper

For the same reasons that MEB has been generalized to MEM, it is natural and desirable to extend SBT to its multicast equivalent. The Minimum Shared

Multicast Tree (SMT) problem is thus at the forefront of this paper, and neither MEB, MEM, nor RAP will be given much attention. Focus is henceforth on instances in which some of the nodes neither initiate new messages nor need to receive messages from others. Utilizing such *non-destination* nodes to forward messages can in many cases contribute to substantial reductions in the total power consumption. Non-destination nodes thus have the potential to play the role of Steiner nodes in the multicast tree. All devices that can initiate a transmission, referred to as *destination* nodes, also have to receive every message.

The remainder of this paper is organized as follows: Section 2 summarizes relevant literature and previous works. Section 3 introduces the notation, explains the network model, and gives a formal definition of the problem. Integer linear programming formulations, valid inequalities and their analysis are presented in Section 4, followed by Section 5, where the models under study are compared with respect to their strength. In Section 6, the models are further extended by variables in higher dimensional spaces, and a constraint generation procedure is developed. A fast method for computing good, but not necessarily optimal solutions is given in Section 7. Computational results are reported in Section 8, and suggestions to future work are summarized in Section 9.

2 Literature Overview

The combinatorial optimization literature is already rich on IP approaches to power minimization in wireless networks. Since the work of Wieselthier et al. (2000), in particular the MEB and MEM problems have been studied extensively. Noteworthy contributions to this research include the IP models studied by Das et al. (2003), Altinkemer et al. (2005), Yuan (2005), Yuan et al. (2008), Bauer et al. (2008), Montemanni and Leggieri (2011) and Leggieri et al. (2008). The latter paper proposes an IP formulation of MEM based on a set covering model, together with a preprocessing procedure reducing the number of constraints. Čagalj et al. (2002) prove that MEB, and thereby MEM, is NP-hard. Numerous contributions to fast, but possibly inexact methods for MEM have thus appeared, and the interested reader is referred to (Hsiao et al. 2013) for an overview. Another study concerning MEM is the work by Guo and Yang (2004), who investigate energy conservation while using adaptive antennas. Unlike omnidirectional antennas, adaptive antennas can concentrate their transmission power in a smaller angle.

Bein and Zheng (2010) study energy efficiency in wireless networks where all nodes can initiate a message, and the communication takes place in a single all-to-all broadcast tree. A heuristic algorithm generating all-to-all broadcast trees is proposed by Bhukya and Singh (2014), along with an experimental evaluation confirming its merit.

The SBT problem has also received some attention in the scientific literature. It was first introduced by Papadimitriou and Georgiadis (2006), who also proved that the problem is NP-hard. In the same article, an approximation algorithm for SBT is developed and analyzed experimentally. Building on this work, Yuan and Haugland (2012) present an integer programming model for SBT, along with an associated decomposition approach.

Simple instances (Yuan and Haugland 2012) show that because of the difference in objective functions, the optimal solutions to RAP and SBT can differ substantially. Kirousis et al. (1997) and Clementi et al. (1999) conduct hardness results for instances of RAP embedded in three- and two-dimensional Euclidean space, respectively. Later, IP models have been developed by e.g. Althaus et al. (2003), Montemanni and Gambardella (2004), Das et al. (2005), and Haugland and Yuan (2011).

Some preliminary work on SMT has recently been published. Ivanova (2016) introduces the problem, and develops an IP model. Moreover, the author studies inexact construction and improvement methods, for which computational results are reported.

3 The Minimum Shared Multicast Tree Problem

Like in studies of other problems in combinatorial optimization, a theoretical investigation of SMT requires abstraction and simplification of the real-world communication network.

3.1 Assumptions and notation

A WANET is modeled by a complete graph $G = (V, E)$, where the set V of nodes represents the set of wireless devices, and the set of edges $E = \{\{i, j\} : i, j \in V, i \neq j\}$ corresponds to the potential links between them. The set $A = \{(i, j) : i, j \in V, \{i, j\} \in E\}$ consists of all arcs that can be derived by directing the edges in E. Nodes in the set $D \subseteq V$ are the destination nodes, also referred to as *source* nodes. For an arbitrary $i \in V$, sets $V \setminus \{i\}$ and $D \setminus \{i\}$ are denoted V_i and D_i, respectively.

Sending a message directly from (to) node i to (from) node j requires a positive *transmission power* $p_{ij} = p_{ji}$. Whenever appropriate, p_{ij} is referred to as the *cost* of $\{i, j\}$, and edges with higher cost than others are said to be *more expensive*. When the nodes are embedded in the plane, p_{ij} is often assumed (Papadimitriou and Georgiadis 2006; Yuan and Haugland 2012; Halgamuge et al. 2009) to be proportional to the Euclidean distance between i and j raised to a power between 2 and 4. For simplicity, we assume for all $i \in V$ that $p_{ij} \neq p_{ik}$ when $j \neq k$, and define $W_{ij} = \{k \in V_i : p_{ik} \geq p_{ij}\}$ as the set of nodes requiring no less power at i than j does. The notation $G' \subseteq G$ means that $G' = (V_{G'}, E_{G'})$ is a subgraph of G, i.e., $E_{G'} \subseteq E$ consists uniquely of edges in E with both end nodes in $V_{G'} \subseteq V$.

Let $y \in \{0,1\}^E$ and $X \in \{0,1\}^{A \times D}$ be binary vectors with components corresponding to edges, and pairs of arcs and destination nodes, respectively. The vector X^s then consists of all components of X corresponding to a pair of which s is the destination. The subgraph of G *induced by* y is defined as $G_y = (V, E_y)$, where $E_y = \{\{i,j\} \in E : y_{ij} = 1\}$. Directed graphs induced by binary vectors over A are defined analogously.

Several integer programming models, all of which exclusively have binary variables, will be considered in this text. If $\mathcal{M}$ is such a model, LP$(\mathcal{M})$ denotes the corresponding continuous relaxation where the domain of each variable is replaced by the interval $[0,1]$. Further, $z\,(\mathcal{M})$ and $z\,(\mathrm{LP}(\mathcal{M}))$ denote the optimal objective function values of the respective models.

3.2 Network model

To illustrate the problem under study, consider the tree depicted in Fig. 1. Seven destination nodes ($D = \{i_1, i_2, a, b, c, d, e\}$) are spanned by use of two Steiner nodes ($V \setminus D = \{i, i_3\}$). Consider node i and its three neighbour nodes i_1, i_2 and i_3, ordered by decreasing power requirement at i. That is, $p_{ii_1} > p_{ii_2} > p_{ii_3}$. If the transmitting node is outside the shaded area (i_2, c, d, or e), then node i relays the signal to two of its neighbors, *including* i_1. Exploiting the wireless advantage, the signal from node i reaches both neighbors if it is assigned power level p_{ii_1}. Otherwise, if the signal source is either i_1, a, or b, node i passes the signal on to neighbors i_2 and i_3. This is accomplished by setting the power level of i to p_{ii_2}.

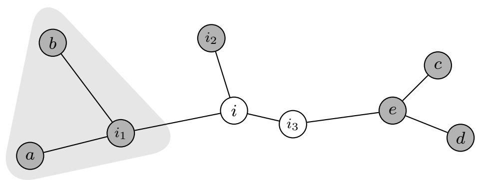


Fig. 1: A multicast tree illustrating the cost contribution from node i to the objective function. Destinations and Steiner nodes are denoted by filled and empty circles, respectively.

In more general terms, let $T \subseteq G$ be a multicast tree, and let $i \in V_T$ be a node with at least two neighbors in T. For multicasting a message, the power to be assigned to node i is either p_{ii_1} or p_{ii_2}, where i_1 and i_2 are neighbors of i such that $p_{ii_1} > p_{ii_2} > p_{ij}$ for all other neighbors $j \neq i_1, i_2$ of i in V_T. If the path from the message source to i intersects i_1, then power p_{ii_2} is sufficient. Otherwise, the power p_{ii_1} is required. At a leaf i with neighbor i_1 in T, the power assignment is p_{ii_1} if i also is the message source, and zero otherwise.

All nodes in D are assumed to be equally likely to initiate a message. The frequency at which power levels p_{ii_1} and p_{ii_2} (let $p_{ii_2} = 0$ if i is a leaf) are required, is thus proportional to the number of sources in D that utilize the respective arc for message forwarding. In the example in Fig. 1, the expected power consumption at node i is thus proportional to $4p_{ii_1} + 3p_{ii_2}$. Summing over all nodes yields the total cost of the tree.

The cost model outlined above aims for minimization of the transmission energy. As pointed out by Halgamuge et al. (2009), transmission constitutes a substantial part of the total energy consumption in wireless sensor networks. The energy consumed by other operations, such as sensing, logging, processing, actuation, and cluster formation is less sensitive to the topology of the multicast tree, and is neglected in the current work. This is in line with the approach taken in the wide range of literature on optimization models for wireless networks mentioned in Section 2. For a detailed energy model for wireless sensor networks, the reader is referred to the article by Halgamuge et al. (2009).

3.3 Problem definition

To express the cost of the multicast tree more rigorously, let $T^s = (V_T, A^s)$ denote the directed tree (arborescence) obtained by directing all edges in T such that they point away from the *root* at s. For each arc $(i, j) \in A$, we let $n_{ij}(T)$ denote the number of destinations s such that (i, j) is the most expensive arc leaving node i in T^s. That is,

$$n_{ij}(T) = \left| \left\{ s \in D : p_{ij} = \max_{k \in V_i : (i,k) \in A^s} p_{ik} \right\} \right|,$$

and the total cost of tree T becomes:

$$c(T) = \sum_{i \in V_T} \sum_{j \in V_T : \{i,j\} \in E_T} p_{ij} n_{ij}(T).$$

The problem under study is thus defined in network optimization terms:

Problem 1 Minimum Shared Multicast Tree: Find a tree $T \subseteq G$ spanning D such that $c(T)$ is minimized.

While the feasible domains of the Minimum Steiner Tree problem and SMT are identical, the cost functions differ. In the former problem, all edges present in the tree contribute to the objective function by a cost that is given by the input data. The input cost factor p_{ij} is, in the case of SMT, scaled by a topology dependent factor $n_{ij}(T)$. Only edges that are the most or second-most expensive edge incident to either of their end nodes get a positive scaling factor. Some edges in T can thus be scaled down to zero cost, while the most expensive ones are scaled up.

Problem 1 also has close resemblance with the SBT problem, which is the special case of SMT where $D = V$. Like many other network design problems, SMT is NP-hard. This follows directly from the NP-hardness of SBT (Papadimitriou and Georgiadis 2006).

In the forthcoming sections, we develop and analyze IP models for SMT. All formulations under study are extensions of models previously developed for the related problems mentioned above.

4 IP Formulations

A basic element of every IP formulation for SMT is a set of constraints modelling a Steiner tree. We investigate two such constraint sets formulated in terms of variables with up to three node indices, and make necessary extensions required by the objective function of SMT. The resulting models are subsequently strengthened by valid inequalities.

4.1 Formulation based on broadcast trees

The first model extends the SBT formulation introduced by Yuan and Haugland (2012) by non-destination nodes, in order to formulate the multicast version of the problem (Ivanova 2016).

4.1.1 Underlying Steiner tree formulation

Feasible solutions to SMT are characterized by a Steiner tree T covering D. Define for each edge $\{i, j\} \in E$ and each destination node $s \in D$, the binary variables $y \in \{0, 1\}^E$ and $X \in \{0, 1\}^{A \times D}$, where $y_{ij} = 1$ iff $\{i, j\} \in E_T$, and $X_{ij}^s = 1$ iff $(i, j) \in A^s$. To induce a graph with an acyclic connected component spanning D, the following constraints are imposed:

$$\sum_{j \in V_i} X_{ji}^s = 1 \qquad i, s \in D, i \neq s, \tag{1a}$$

$$\sum_{j \in V_i} X_{ji}^s \leq 1 \qquad i \in V \setminus D, s \in D, \tag{1b}$$

$$X_{ij}^s \leq \sum_{k \in V_i \setminus \{j\}} X_{ki}^s \qquad i \in V \setminus D, (i, j) \in A, s \in D, \tag{1c}$$

$$X_{ij}^s + X_{ji}^s = y_{ij} \qquad \{i, j\} \in E, s \in D, \tag{1d}$$

$$X_{is}^s = 0 \qquad s \in D, (i, s) \in A, \tag{1e}$$

$$y \in \{0, 1\}^E, X \in \{0, 1\}^{A \times D}. \tag{1f}$$

Constraints (1a) ensure that all destination nodes but s have a unique entering arc in T^s. Inequalities (1b)–(1c) say that non-destination nodes do not have more than one entering arc, and that they do have one if they have leaving arcs. Constraints (1d) state that the arc set of T^s consists of directed versions of the edge set of T, and (1e) ensure that s is the root of T^s. More formally, the effect of the constraints can be stated as follows:

Lemma 1 *If (X, y) satisfies (1a)–(1f), then G_y has a tree spanning D as one of its connected components. Conversely, for any tree $T \subseteq G$ spanning D, there exists a pair (X, y) satisfying (1a)–(1f) such that T is a connected component of G_y.*

Proof By (1e) and (1a)–(1b), the in-degree of $s \in D$ is 0 in G_{X^s}, whereas it is at most one at all other nodes. Hence, the connected components of G_{X^s} are directed cycles, arborescences, and isolated nodes, and one of the components is an arborescence rooted at s. Constraints (1d) imply that G_y consists of undirected versions of the same components, showing that no cycle in G_y contains destination nodes. Let $t \in D$, and let P be the maximal path in G_{X^s} that terminates in t. Existence and uniqueness of this path are assured since t belongs to an acyclic connected component. According to (1c), each non-destination in P has an entering arc in P, while constraints (1a) ensure that the same is true for every destination node, except s. It follows that the first node of P is s. Thus, destinations s and t belong to a joint, acyclic connected component of G_y.

The second claim of the lemma is obvious. □

4.1.2 SMT model [$\mathcal{X}_1$]

To reflect that only expensive edges in T contribute to the objective function of Problem 1, we introduce the variables:

$$\pi_{ij}^s = \begin{cases} 1 & \text{if } (i,j) \text{ is the most expensive arc leaving node } i \text{ in } T^s, \\ 0 & \text{otherwise,} \end{cases}$$

leading to the following formulation $\mathcal{X}_1$ of SMT:

$$\min \sum_{(i,j)\in A} p_{ij} \sum_{s\in D} \pi_{ij}^s, \tag{1g}$$

$$\text{s.t.}$$

$$(1a) - (1f),$$

$$X_{ij}^s \le \sum_{k\in W_{ij}} \pi_{ik}^s \qquad s \in D, (i,j) \in A, \tag{1h}$$

$$\pi \in \{0,1\}^{A\times D}. \tag{1i}$$

Constraints (1h) say that if $(i, j) \in A_{T^s}$, then the cost of the most expensive arc in T^s leaving i is at least p_{ij}. Since $\sum_{s\in D} \pi_{ij}^s = n_{ij}(T)$ is the number of arborescences in which (i,j) is the most expensive arc leaving i, (1g) thus agrees with the objective of Problem 1.

Proposition 1 *If (X,y,π) is an optimal solution to $\mathcal{X}_1$, then G_y consists of isolated non-destination nodes and a tree that solves Problem 1.*

Proof Follows directly from $p > 0$ and Lemma 1. □

Model $\mathcal{X}_1$ is a slightly modified version of the SMT model introduced by Ivanova (2016), which contains a set of constraints disallowing non-destination leaves, and a weaker version of constraints (1c). Formulation (1g)–(1i) is *minimal*, in the sense that each set of constraints is necessary in order to give a valid formulation. In the next section, we strengthen the formulation by adding redundant valid inequalities.

4.1.3 Valid inequalities [$\mathcal{X}_2$]

Adding the following valid inequalities to $\mathcal{X}_1$ gives another SMT-model, denoted $\mathcal{X}_2$:

$$\sum_{j\in V_i} X^s_{ji} \le \sum_{j\in V_i} X^s_{ij} \qquad i \in V\setminus D, s\in D, \tag{1j}$$

$$\sum_{j\in V_s} \pi^s_{sj} = 1 \qquad s \in D, \tag{1k}$$

$$\sum_{j\in V_i\setminus\{s\}} \pi^s_{ij} = \sum_{j\in V_i} X^s_{ji} \qquad i \in V\setminus D, s\in D. \tag{1l}$$

Constraints (1j) ensure that the number of arcs leaving a non-destination in T^s is no less than the number of entering arcs. Their effect is to disallow non-destination leaves. Equations (1k) say that in arborescence T^s, exactly one arc (s,j) leaving the root s is declared to be the largest. Validity of the constraints follows immediately because the root has at least one leaving arc. As constraints (1l) state, if an arc enters non-destination node i in T^s, then i also has at least one out-going arc, exactly one of which is the most expensive. The root s is excluded from the summation set on the left of (1l) since no arcs in T^s enter s.

Remark 1 The sets (1j)–(1l) of valid inequalities are *independent* in the following sense: There exist instances where the removal of either set implies that $z(\mathrm{LP}(\mathcal{X}_2))$ takes a smaller value. This includes the instance where $|V| = 9$, $|D| = 5$, all power requirements (arc costs) p_{ij} equal the square of the Euclidean distance between nodes i and j, and the nodes are located in the plane with respective coordinates (destinations mentioned first) (47, 65), (8, 65), (26, 12), (78, 27), (8, 8), (73, 66), (13, 55), (91, 44), and (76, 74).

4.2 Formulation based on network flows

There are many formulations for the Minimum Steiner Tree problem (Goemans and Myung 1993) that can serve as a basis for modelling SMT. We consider the multi-commodity network flow model developed by Polzin and Daneshmand (2001) (denoted P_F in their work). A given destination $s_0 \in D$ plays a particular role in the models. For the sake of simplified notation, s_0 as subscript (superscript) is henceforth replaced by subscript (superscript) 0.

4.2.1 Underlying Steiner tree formulation

To identify a Steiner arborescence $T^0 = (V^0, A^0)$ rooted at s_0, we define variables $g \in \{0,1\}^A$ and $F \in \{0,1\}^{A \times D_0}$, where $g_{ij} = 1$ iff $(i,j) \in A$, and $F^s_{ij} = 1$ iff the path in T^0 from s_0 to destination s contains arc (i,j). Variables g_{ij} and F^s_{ij} are traditionally referred to as the *design variable* of arc (i,j) and the *flow of commodity* s along (i,j), respectively. The constraints of the *multi-commodity flow formulation* of the Minimum Steiner Tree problem are:

$$F^s_{ij} \leq g_{ij} \qquad s \in D_0, (i,j) \in A, \tag{2a}$$

$$\sum_{j \in V_i} F^s_{ji} - \sum_{j \in V_i} F^s_{ij} = \begin{cases} 1, & i = s, \\ 0, & i \in V \setminus \{s_0, s\}, \end{cases} \qquad s \in D_0, \tag{2b}$$

$$\sum_{j \in V_i} g_{ji} \leq 1 \qquad i \in V \setminus D, \tag{2c}$$

$$F^s_{si} = 0 \qquad s \in D_0, i \in V_s, \tag{2d}$$

$$F^s_{is} = g_{is} \qquad s \in D_0, i \in V_s, \tag{2e}$$

$$g_{i0} = 0 \qquad i \in V_0, \tag{2f}$$

$$g \in \{0,1\}^A, F \in \{0,1\}^{A \times D_0}. \tag{2g}$$

The *flow conservation* constraints (2b) ensure that T^0 is connected, and the *capacity* constraints (2a) state that flow of any commodity is allowed only if $(i,j) \in A^0$. Multi-commodity flow formulations appear abundantly in the network design literature, and proof of the following claim is therefore omitted:

Lemma 2 *Assume (F, g) satisfies (2a)–(2g). Then, some connected component $T^0 \subseteq G_g$ is an s_0-rooted arborescence spanning D, and some connected component of $G_{F^s} \subseteq T^0$ is a path from s_0 to $s \in D_0$. Conversely, for any s_0-rooted arborescence T^0 spanning D, there exists a pair (F, g) satisfying (2a)–(2g) such that T^0 is a connected component of G_g.*

To model the objective function of SMT, the following result is useful:

Proposition 2 *Assume (F,g) satisfies (2a)–(2g), and let $T^s = (V^s, A^s)$ be the arborescence obtained by redirecting the arcs in T^0 such that they point away from $s \in D_0$. Then,*

$$g_{ij} - F^s_{ij} + F^s_{ji} = \begin{cases} 1, & \text{if } (i,j) \in A^s, \\ 0, & \text{otherwise.} \end{cases}$$

Proof Because T^0 is acyclic, $g_{ij} + g_{ji} \leq 1$, and it follows from the capacity constraints (2a) that $g_{ij} + F^s_{ji} \leq 1$. The same constraints also imply $g_{ij} - F^s_{ij} \geq 0$. Hence, $g_{ij} - F^s_{ij} + F^s_{ji} \in \{0,1\}$.

Arborescence T^s contains arc (i,j) iff (i,j) is in A^0 but not on a path to s, or its reverse arc (j,i) is on such a path. By Lemma 2, this disjunction is equivalent to $g_{ij} - F^s_{ij} = 1$ or $F^s_{ji} = 1$. The result then follows since the two conditions are mutually exclusive. □

4.2.2 SMT model [$\mathcal{F}_1$]

Recall from Section 4.1.2 that π^s_{ij} determines whether (i,j) is the most expensive arc leaving node i in arborescence T^s. Taking advantage of Prop. 2, we can thus formulate Problem 1 in terms of variables F, g, and π, resulting in model $\mathcal{F}_1$:

$$\min \sum_{(i,j)\in A} p_{ij} \sum_{s \in D} \pi^s_{ij},$$

s.t.

$$(2a) - (2g),$$

$$g_{ij} - F^s_{ij} + F^s_{ji} \leq \sum_{k \in W_{ij}} \pi^s_{ik} \qquad s \in D_0, (i,j) \in A, \tag{2h}$$

$$g_{ij} \leq \sum_{k \in W_{ij}} \pi^0_{ik} \qquad (i,j) \in A, \tag{2i}$$

$$\pi \in \{0,1\}^{A \times D}. \tag{2j}$$

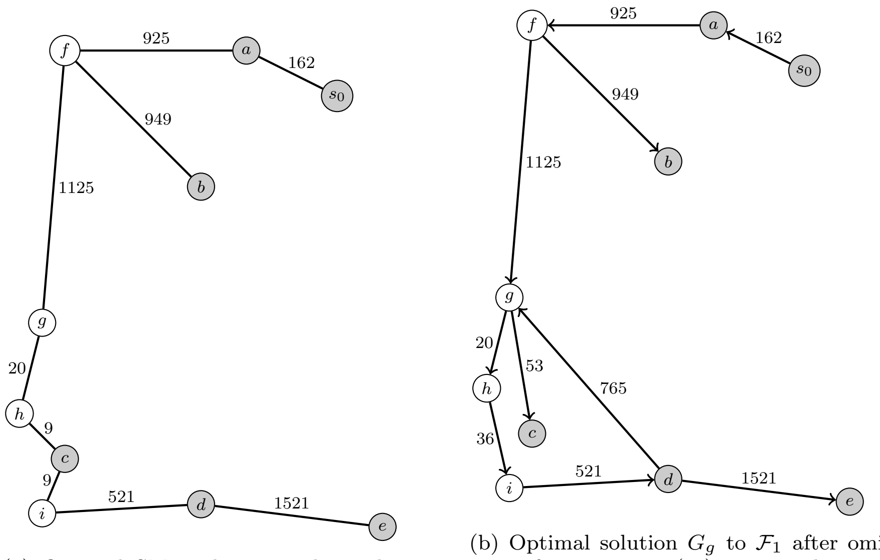
Like (1h), constraints (2h) ensure that if $(i,j) \in A^s$, then $\pi^s_{ik} = 1$ for some arc (i,k) at least as expensive as (i,j). As g_{ij} determines whether $(i,j) \in A^0$, the corresponding constraint for destination s_0 is simplified to (2i). Validity of $\mathcal{F}_1$ is thus stated as a corollary to Prop. 2:

Corollary 1 *If (F,g,π) is an optimal solution to $\mathcal{F}_1$, then G_g consists of isolated non-destination nodes and an arborescence T^0. The tree obtained by disregarding arc directions in T^0 is an optimal solution to Problem 1.*

Remark 2 Like $\mathcal{X}_1$, also $\mathcal{F}_1$ is a minimal formulation. Constraints (2c) prevent non-destinations from having multiple entering arcs. In a Minimum Steiner Tree formulation, where a linear function of g is minimized subject to constraints (2a)–(2b) and (2d)–(2g), constraints (2c) are implied by optimality.

The necessity of (2c) in formulation $\mathcal{F}_1$ of SMT is demonstrated by the optimal solution to $\mathcal{F}_1$ after removal of (2c) in the following instance: $V = \{s_0, a, \dots, i\}$ with coordinates (3, 61), (79, 80), (32, 54), (93, 21), (12, 70), (93, 60), (72, 78), (76, 80), (39, 84), and (82, 80), respectively, edge costs equal to the square of the Euclidean distance between end points, and $D = \{s_0, a, \dots, f\}$. Figure 2 depicts the optimal solution to SMT, and the solution to $\mathcal{F}_1$ after removal of (2c).

Remark 3 The obviously valid inequalities (2d)–(2e) are not necessary in the Minimum Steiner Tree formulation, but have to be included in the formulation $\mathcal{F}_1$ of SMT in order to disallow nodes in D_0 to have multiple entering arcs. Removal of (2d) or (2e) implies that the optimal solution to the instance where $V = \{s_0, a, \dots, k\}$ with coordinates (43, 15), (89, 88), (56, 12), (44, 66), (60, 42), (91, 48), (64, 8), (64, 34), (68, 9), (41, 53), (85, 67), and (93, 49), respectively, edge costs defined as in Remark 2, and $D = \{s_0, a, \dots, h\}$, is not a feasible solution to SMT.



(a) Optimal SMT-solution with total cost 25156.

(b) Optimal solution G_g to $\mathcal{F}_1$ after omission of constraints (2c). Its total cost is 25148, but the cycle shows that it is infeasible in SMT.

Fig. 2: Instance demonstrating necessity of constraints (2c) in $\mathcal{F}_1$. Edge labels denote costs. For better legibility, some edges are contracted in the illustration.

4.2.3 Valid inequalities [$\mathcal{F}_2$]

Valid inequalities analogous to those leading to model $\mathcal{X}_2$ in Section 4.1.3 are added to $\mathcal{F}_1$, resulting in a model denoted $\mathcal{F}_2$. Analogously to (1j), to enforce non-destination nodes included in the spanning arborescence to have at least one child node, the following constraints are introduced (Polzin and Daneshmand 2001):

$$\sum_{j\in V_i} g_{ji} - \sum_{j\in V_i} g_{ij} \le 0 \qquad\qquad i \in V \setminus D. \tag{2k}$$

According to Prop. 2, if $i \in V \setminus D$ and $g_{ji} - F^s_{ji} + F^s_{ij} = 1$ for some $j \in V_i$ and $s \in D_0$, then i is a Steiner node used to establish connection between s and some other destination. Since i also has a leaving arc in T^s, we have $\pi^s_{ik} = 1$ for some $k \neq i, s$. This proves validity of the constraint $\sum_{j\in V_i} \left(g_{ji} - F^s_{ji} + F^s_{ij}\right) = \sum_{j\in V_i\setminus\{s\}} \pi^s_{ij}$. Application of flow conservation (2b) yields the valid constraints

$$\sum_{j\in V_i\setminus\{s\}} \pi^s_{ij} = \sum_{j\in V_i} g_{ji} \qquad i \in V \setminus D, s \in D. \tag{2l}$$

Finally, constraints (1k) are valid, completing formulation $\mathcal{F}_2$ as model $\mathcal{F}_1$ extended by (2k)–(2l) and (1k).

Remark 4 With s_0 equal to the destination node in position (47, 65), the instance given in Remark 1 also proves independence of the three sets (1k), (2k)–(2l) of valid inequalities.

5 Relations Between the Models

In this section, we analyze and compare the lower bounds provided by optimal solutions with the continuous relaxations of the models formulated in Section 4. For any integer programming models $\mathcal{M}$ and $\mathcal{M}'$ of SMT, we say that LP($\mathcal{M}$) is *at least as strong as* LP($\mathcal{M}'$) if $z\,(\text{LP}(\mathcal{M})) \ge z\,(\text{LP}(\mathcal{M}'))$ for all SMT-instances, and that it is *stronger* if there also exists an instance in which the inequality is strict.

5.1 Comparing $\mathcal{X}_1$ and $\mathcal{F}_1$

For all $(i, j) \in A$, define $F^*_{ij} = \max_{s\in D_0} F^s_{ij}$. To prove that LP($\mathcal{F}_1$) is at least as strong as LP($\mathcal{X}_1$), the following claim is instrumental.

Lemma 3 *All feasible solutions (F, g, π) to LP($\mathcal{F}_1$) satisfy $\sum_{k\in V_i} g_{ki} = 1$ $(i \in D_0)$ and $\sum_{k\in V_i} g_{ki} \le 1$ $(i \in V \setminus D)$.*

Proof Let $i \in D_0$. Utilizing first (2e), next flow conservation (2b) at $i = s$, and finally (2d), we get $\sum_{j\in V_i} g_{ji} = \sum_{j\in V_i} F^i_{ji} = 1 + \sum_{j\in V_i} F^i_{ij} = 1$. For $i \in V \setminus D$, the claim is identical to (2c). $\square$

Lemma 4 *In all SMT-instances, there exists an optimal solution* (F, g, π) *to* LP($\mathcal{F}_1$) *satisfying*

$$\min\{F^s_{ij}, F^s_{ji}\} = 0 \qquad \{i,j\} \in E, s \in D_0, \textit{ and} \tag{3a}$$
$$g_{ij} = F^*_{ij} \qquad (i,j) \in A. \tag{3b}$$

Proof Obviously, an optimal solution to LP($\mathcal{F}_1$) exists, so let $(\bar{F}, \bar{g}, \pi)$ denote one such solution. For all $(i,j) \in A$ and $s \in D_0$, let $F^s_{ij} = \bar{F}^s_{ij} - \min\{\bar{F}^s_{ij}, \bar{F}^s_{ji}\}$, and let g be chosen according to (3b). Consequently, (3a)–(3b) are satisfied, and the objective function values of (F, g, π) and $(\bar{F}, \bar{g}, \pi)$ are equal.

To prove optimality of (F, g, π), it suffices to prove feasibility. The capacity constraints (2a) follow directly from (3b). Because $\bar{F}$ satisfies the flow conservation constraints (2b), also F does, since $F^s_{ji} - F^s_{ij} = \bar{F}^s_{ji} - \bar{F}^s_{ij}$. Further, $g_{ij} = F^*_{ij} \leq \bar{F}^*_{ij} \leq \bar{g}_{ij}$, proving that also (2c) and (2h)–(2i) are satisfied. Finally, (2d)–(2e) follow directly from $\bar{F}^s_{si} = 0$ and $\bar{F}^s_{is} = \bar{g}_{is}$, and (2f) follow from $g_{i0} \leq \bar{g}_{i0} = 0$. Thus, (F, g, π) is feasible in LP($\mathcal{F}_1$). □

Lemma 5 *If* (F, g, π) *is feasible in* LP($\mathcal{F}_1$) *with (3a)–(3b) satisfied, then*

$$g_{ij} + g_{ji} \leq 1 \quad \textit{for all } \{i,j\} \in E. \tag{4}$$

Proof Assume $g_{ij} + g_{ji} > 1$ for some $\{i,j\} \in E$. Because $g_{i0} = 0$, this implies $i \neq s_0 \neq j$. By (3b), there exist $s, t \in D_0$ such that $F^s_{ij} + F^t_{ji} > 1$. According to (3a), $F^s_{ij} > 1 - F^t_{ji} \geq 0$ shows that $F^s_{ji} = 0$. Flow conservation at node i thus implies $1 < F^s_{ij} + F^t_{ji} \leq \sum_{k \in V_i \setminus \{j\}} F^s_{ki} + F^t_{ji} \leq \sum_{k \in V_i \setminus \{j\}} g_{ki} + g_{ji}$, contradicting Lemma 3. □

Proposition 3 LP($\mathcal{F}_1$) *is at least as strong as* LP($\mathcal{X}_1$).

Proof Consider an optimal solution (F, g, π) to LP($\mathcal{F}_1$) satisfying the conditions of Lemma 4. We prove that there exist corresponding $X \in [0,1]^{A \times D}$ and $y \in [0,1]^E$ such that (X, y, π) is a feasible solution to LP($\mathcal{X}_1$). Let

$$X^0_{ij} = g_{ij} \qquad (i,j) \in A, \tag{5a}$$
$$X^s_{ij} = g_{ij} + F^s_{ji} - F^s_{ij} \qquad (i,j) \in A, s \in D_0, \tag{5b}$$
$$y_{ij} = g_{ij} + g_{ji} \qquad \{i,j\} \in E. \tag{5c}$$

For distinct destinations $i, s \in D_0$, Lemma 3 yields

$$\sum_{j \in V_i} X^s_{ji} = \sum_{j \in V_i} \left(g_{ji} + F^s_{ij} - F^s_{ji}\right) = \sum_{j \in V_i} g_{ji} = 1,$$

proving that (1a) is satisfied for $i \neq s_0 \neq s$. By applying (2f), we get for $i = s_0 \neq s$, that $\sum_{j \in V_i} X^s_{ji} = \sum_{j \in V_0} \left(g_{j0} + F^s_{0j} - F^s_{j0}\right) = \sum_{j \in V_0} F^s_{0j} = 1$, where the latter identity is obtained by summing all flow conservation constraints (2b) in which variables are superscripted by destination s. Likewise, for $i \neq s_0 = s$, we get $\sum_{j \in V_i} X^s_{ji} = \sum_{j \in V_i} g_{ji} = 1$. Hence, X satisfies (1a).

Also for $i \in V \setminus D$, we have $\sum_{j \in V_i} X^s_{ji} = \sum_{j \in V_i} g_{ji}$, and constraints (1b) follow from (2c). Constraints (1d) follow directly from the choice of values of X and y, and (1e) follows from (2d)–(2e). Further, $y_{ij} \in [0,1]$ holds according to Lemma 5, and $X^s_{ij} \in [0,1]$ follows from the capacity constraint (2a).

It remains to prove (1c). First, flow conservation at node i is used to show

$$X^s_{ij} - \sum_{k \in V_i \setminus \{j\}} X^s_{ki} = g_{ij} + F^s_{ji} - F^s_{ij} - \sum_{k \in V_i \setminus \{j\}} (g_{ki} + F^s_{ik} - F^s_{ki}) = g_{ij} - \sum_{k \in V_i \setminus \{j\}} g_{ki}.$$

It follows from condition (3b) in Lemma 4 that for some $t \in D_0$,

$$X^s_{ij} - \sum_{k \in V_i \setminus \{j\}} X^s_{ki} = F^t_{ij} - \sum_{k \in V_i \setminus \{j\}} g_{ki} \le F^t_{ij} - \sum_{k \in V_i \setminus \{j\}} F^t_{ki}.$$

If $F^t_{ij} = 0$, the right hand side of the inequality is obviously non-positive. To prove the same when $F^t_{ij} > 0$, observe that (3a) implies $F^t_{ji} = 0$, which yields

$$F^t_{ij} - \sum_{k \in V_i \setminus \{j\}} F^t_{ki} = F^t_{ij} - \sum_{k \in V_i} F^t_{ki} \le \sum_{k \in V_i} F^t_{ik} - \sum_{k \in V_i} F^t_{ki} = 0.$$

Since the objective function values of (F, g, π) and (X, y, π) are identical in their respective relaxations, the proof is complete. □

Remark 5 Section 8.1 reports instances proving that LP($\mathcal{F}_1$) is *stronger than* LP($\mathcal{X}_1$).

5.2 Comparing $\mathcal{X}_2$ and $\mathcal{F}_2$

For all $(i, j) \in A$ where $i, j \in V \setminus D$, let

$$\sigma_{ij}(F, g) = \min \left\{ g_{ij} - F^*_{ij}, \sum_{k \in V_i} (g_{ik} - g_{ki}) \right\}.$$

That is, $\sigma_{ij}(F, g)$ is the smallest among slacks in constraints (2a) at arc (i, j) and constraint (2k) at node i.

Lemma 6 *In all SMT-instances, there exists an optimal solution (F, g, π) to LP($\mathcal{F}_2$) satisfying (3a),*

$$\min\left\{g_{ij} - F^*_{ij}, g_{ji} - F^*_{ji}\right\} = 0 \qquad \{i,j\} \in E, \tag{6a}$$

$$g_{ij} = F^*_{ij} \qquad (i,j) \in A : \{i,j\} \cap D \neq \emptyset, \text{ and} \tag{6b}$$

$$\sigma_{ij}(F, g) = 0 \qquad (i,j) \in A : \{i,j\} \subseteq V \setminus D. \tag{6c}$$

Proof By following the initial step of the proof of Lemma 4, it is shown that $\mathrm{LP}(\mathcal{F}_2)$ has an optimal solution $(F, \bar{g}, \pi)$ satisfying (3a). For all $(i, j) \in A$, let $\delta_{ij} = \min\left\{\bar{g}_{ij} - F^*_{ij}, \bar{g}_{ji} - F^*_{ji}\right\}$, and let $g_{ij} = \bar{g}_{ij} - \delta_{ij}$. Hence, (F, g) satisfies (6a). By exploiting $g \leq \bar{g}$, $g_{ij} - g_{ji} = \bar{g}_{ij} - \bar{g}_{ji}$, and feasibility of $(F, \bar{g}, \pi)$, it is seen straightforwardly that also (F, g, π) is feasible.

For $j \in D_0$ (for $j = s_0$), (6b) follows directly from (2e) (from (2f)). Consider an $i \in D$ and $j \in V_i \setminus D$ such that $g_{ij} > F^*_{ij}$. Reducing g_{ij} to F^*_{ij}, while keeping all other variables unchanged, preserves feasibility and the objective function value. Thus, (F, g) is henceforth assumed to satisfy (6b).

To prove that also (6c) can be satisfied, consider the *slack-induced digraph* $H[F, g] = (V \setminus D, A[F, g])$ of non-destination nodes, where $A[F, g] = \left\{(i, j) \in A : i, j \in V \setminus D, F^*_{ij} < g_{ij}\right\}$. Assume $C = (i_0, \ldots, i_{c-1}, i_c)$ is a cycle of length c in H ($i_c = i_0$), and let $\varepsilon = \min\left\{g_{i_l i_{l+1}} - F^*_{i_l i_{l+1}} : l = 0, \ldots, c-1\right\} > 0$. Define $g'_{i_l i_{l+1}} = g_{i_l i_{l+1}} - \varepsilon$ ($l = 0, \ldots, c-1$), and $g'_{ij} = g_{ij}$ for arcs (i, j) not in C. Because $\sum_{j \in V_i} g'_{ji} - \sum_{j \in V_i} g'_{ij} = \sum_{j \in V_i} g_{ji} - \sum_{j \in V_i} g_{ij}$, g' satisfies (2k). Since $g' \leq g$, it follows that (F, g', π) is feasible in $\mathrm{LP}(\mathcal{F}_2)$, with the same objective function value as (F, g, π). Then, $A[F, g'] \subset A[F, g]$, and for some $l = 0, \ldots, c-1$, $(i_l, i_{l+1}) \in A[F, g] \setminus A[F, g']$. An induction argument hence proves the existence of an optimal solution to $\mathrm{LP}(\mathcal{F}_2)$ for which the corresponding slack-induced digraph is acyclic. It is henceforth assumed that (F, g, π) is such a solution.

Assume arc (j_0, j_1) violates (6c). We show that the capacity can be reduced to $\tilde{g} \leq g$ such that

- $(F, \tilde{g}, \pi)$ is feasible in $\mathrm{LP}(\mathcal{F}_2)$,
- $\sigma_{ij}(F, \tilde{g}) > 0$ only for arcs (i, j) where $\sigma_{ij}(F, g) > 0$, and
- either $\sigma_{j_0 j_1}(F, \tilde{g}) = 0$ or $A[F, \tilde{g}] \subsetneq A[F, g]$.

Finitely many reductions, and replacements of g by $\tilde{g}$, are then sufficient to achieve $\sigma_{j_0 j_1}(F, g) = 0$, which completes the proof.

Let $(j_1, \ldots, j_p)$ be a path with $p \geq 1$ nodes in $H[F, g]$, such that j_p has no out-neighbors in $H[F, g]$. Node j_p exists since the digraph is acyclic. Define the maximum capacity reduction as

$$\sigma^* = \min\left\{\sigma_{j_0 j_1}(F, g), \min\left\{g_{j_l, j_{l+1}} - F^*_{j_l, j_{l+1}} : l = 1, \ldots, p-1\right\}\right\} > 0.$$

Let $\tilde{g}_{j_l j_{l+1}} = g_{j_l j_{l+1}} - \sigma^*$ for $l = 0, 1, \ldots, p-1$, and let $\tilde{g}_{ij} = g_{ij}$ for all other arcs $(i, j) \in A$. Then, $(F, \tilde{g}, \pi)$ is feasible since, in comparison to (F, g, π), the left hand side of inequality (2k) is increased by no more than the current slack at node $i = j_0$, reduced at node $i = j_p$, and unchanged at all other nodes $i \in V \setminus D$. For all arcs (i, j) between non-destination nodes, $\sigma_{ij}(F, \tilde{g}) \leq \sigma_{ij}(F, g)$, except from arcs for which $i = j_p$. But since $(j_p, j) \notin A[F, g]$, we have that $\sigma_{j_p j}(F, \tilde{g}) \leq \tilde{g}_{j_p j} - F^*_{j_p j} = g_{j_p j} - F^*_{j_p j} = 0$. It is finally observed that $\sigma_{j_0 j_1}(F, \tilde{g}) = 0$ if $\sigma^* = \sigma_{j_0 j_1}(F, g)$, and that $(j_l, j_{l+1}) \in A[F, g] \setminus A[F, \tilde{g}]$ if $\sigma^* = g_{j_l, j_{l+1}} - F^*_{j_l, j_{l+1}}$ for some $l = 1, \ldots, p-1$. $\square$

Lemma 7 *If (F, g, π) is a feasible solution to LP($\mathcal{F}_2$) satisfying the conditions in Lemma 6, then (4) is satisfied. Moreover,*

$$g_{ij} \leq \sum_{k \in V_i \setminus \{j\}} g_{ki} \quad (i,j) \in A. \tag{7}$$

Proof To prove (7), we only consider arcs (i, j) where $g_{ij} > 0$, as the inequality to be proved obviously holds otherwise.

Assume first that $g_{ij} = F^*_{ij}$. Then, there exists a destination node $t \in D_0 \setminus \{i\}$ such that $F^t_{ij} = g_{ij}$. Because (3a) gives $F^t_{ji} = 0$, flow conservation (2b) at node i implies $g_{ij} = F^t_{ij} \leq \sum_{k \in V_i \setminus \{j\}} F^t_{ki} \leq \sum_{k \in V_i \setminus \{j\}} g_{ki}$.

Assume next that $g_{ij} > F^*_{ij}$. Conditions (6a) and (6b) of Lemma 6 imply, respectively, $g_{ji} = F^*_{ji}$ and $i, j \in V \setminus D$. Because $g_{ij} > F^*_{ij}$, condition (6c) of Lemma 6 gives $0 = \sigma_{ij}(F, g) = \sum_{k \in V_i} (g_{ik} - g_{ki})$. Thus,

$$g_{ij} = g_{ji} + \sum_{k \in V_i \setminus \{j\}} (g_{ki} - g_{ik}) \,. \tag{8}$$

If $g_{ji} = 0$, (7) follows directly from (8). Otherwise, there exists a $t \in D_0$ such that $g_{ji} = F^t_{ji}$ and $F^t_{ij} = 0$. Flow conservation at i then shows $g_{ji} = F^t_{ji} \leq \sum_{k \in V_i \setminus \{j\}} F^t_{ik} \leq \sum_{k \in V_i} g_{ik}$. Adding this inequality to (8) proves (7).

To prove (4), assume $g_{ij} + g_{ji} > 1$ for some $\{i, j\} \in E$. By (7), this implies $1 < g_{ij} + g_{ji} \leq \sum_{k \in V_i} g_{ki}$, contradicting Lemma 3. □

Proposition 4 *LP($\mathcal{F}_2$) is at least as strong as LP($\mathcal{X}_2$).*

Proof Consider an optimal solution (F, g, π) to LP($\mathcal{F}_2$) satisfying the conditions of Lemma 6. We prove that (X, y, π) as defined by (5a)–(5c) is a feasible solution to LP($\mathcal{X}_2$).

The proof that (X, y, π) satisfies constraints (1a)–(1b), (1d)–(1e), and (1h), follows the proof of Prop. 3. To prove (1c), let $i \in V \setminus D$, $j \in V_i$, and $s \in D$. Then, transformations (5a)–(5b) yield $X^s_{ij} - \sum_{k \in V_i \setminus \{j\}} X^s_{ki} = g_{ij} - \sum_{k \in V_i \setminus \{j\}} g_{ki}$, and (1c) follows from (7) in Lemma 7. Because Lemma 7 also states that $g_{ij} + g_{ji} \leq 1$, we further have that $X^s_{ij} \in [0, 1]$ and $y_{ij} \in [0, 1]$.

Constraints (1j) hold because $\sum_{j \in V_i} \left(X^s_{ji} - X^s_{ij} \right) = \sum_{j \in V_i} (g_{ji} - g_{ij}) \leq 0$ according to constraints (2k). Finally, (2l) implies $\sum_{j \in V_i} X^s_{ji} - \sum_{j \in V_i \setminus \{s\}} \pi^s_{ij} = \sum_{j \in V_i} g_{ji} - \sum_{j \in V_i \setminus \{s\}} \pi^s_{ij} = 0$, which proves that (X, π) satisfies (1l). □

Remark 6 Section 8.1 reports instances proving that LP($\mathcal{F}_2$) is *stronger than* LP($\mathcal{X}_2$).

6 Stronger Formulations

Aiming for stronger LP bounds than those provided by the relaxations of the models introduced in Section 4, this section is devoted to extensions of $\mathcal{X}_2$ and $\mathcal{F}_2$. Variables with four node indices are added to both models. Constraints connecting new and original variables then contribute to the improved bounds.

6.1 Extension of $\mathcal{X}_2$ by destination-to-destination variables [$\mathcal{X}_3$]

Let $A(D) = A \cap (D \times D)$ be the set of ordered pairs of distinct destination nodes. For a tree T solving Problem 1, define the variables $x \in \{0,1\}^{A \times A(D)}$, where $x^{st}_{ij} = 1$ iff $(i,j) \in A$ is an arc on the path from s to t in arborescence T^s. Model $\mathcal{X}_2$ is strengthened by constraints on x, which yields model $\mathcal{X}_3$:

$$\min \sum_{(i,j)\in A} p_{ij} \sum_{s\in D} \pi^s_{ij},$$

$$\text{s.t.}$$

$$(1a) - (1f) \text{ and } (1h) - (1l),$$

$$\sum_{j\in V_i} x^{st}_{ij} - \sum_{j\in V_i} x^{st}_{ji} = 0 \qquad (s,t) \in A(D), i \in V \setminus \{s,t\}, \tag{6d}$$

$$\sum_{j\in V_t} x^{st}_{jt} = 1 \qquad (s,t) \in A(D), \tag{6e}$$

$$x^{st}_{ij} \le X^s_{ij} \qquad (i,j) \in A, (s,t) \in A(D), \tag{6f}$$

$$x^{st}_{ij} = x^{ts}_{ji} \qquad (i,j) \in A, (s,t) \in A(D), \tag{6g}$$

$$\sum_{k\in W_{ij}} x^{st}_{ik} \le \sum_{k\in W_{ij}} \pi^s_{jk} \qquad (i,j) \in A, (s,t) \in A(D), \tag{6h}$$

$$x \in \{0,1\}^{A\times A(D)}. \tag{6i}$$

Variable x^{st}_{ij} is also interpreted as the *flow of commodity* (s,t) along arc (i,j). Correspondingly, (6d)–(6e) can be viewed as flow conservation constraints, stating that there exists a path from s to t in T^s. Next, constraints (6f) express the obvious relation between X and x, that (i,j) is an arc on the path from s to t only if it exists in T^s. The flow symmetry equations (6g) state that the path from s to t contains (i,j) iff the reverse path contains the reverse arc (j,i). Finally, constraints (6h) reflect the fact that if the path from s to t intersects (i,j), then the most expensive arc leaving i in T^s has cost p_{ij} or more.

6.2 Extension of $\mathcal{F}_2$ by variables of joint flow [$\mathcal{F}_3$]

Similarly to the extension $\mathcal{X}_3$ of $\mathcal{X}_2$ by x-variables, the model $\mathcal{F}_2$ is extended by variables with four node indices. Analogously to $A(D)$, let $E(D_0) = E \cap (D_0 \times D_0)$ be the set of *unordered* pairs of destination nodes other than s_0. Define variables $f \in \{0,1\}^{A\times E(D)}$, where $f^{st}_{ij} = f^{ts}_{ji}$ takes the value 1 iff the path in T^0 from s_0 to s intersects with the path from s_0 to t at arc (i,j). Variables f^{st}_{ij} can also be regarded as the *joint flow* from s_0 to s and t (Polzin and Daneshmand 2001). In the cited work, they are successfully used to strengthen formulations for the Minimum Steiner Tree problem. The new variables enable formulation of an extended model denoted $\mathcal{F}_3$:

$$\min \sum_{(i,j)\in A} p_{ij} \sum_{s\in D} \pi_{ij}^s,$$

$$\text{s.t.}$$

$$(2b) - (2l) \text{ and } (1k),$$

$$\sum_{j\in V_i} \left(f_{ij}^{st} - f_{ji}^{st}\right) \leq \begin{cases} 1, & i = s_0, \\ 0, & i \in V_0, \end{cases} \quad \{s,t\} \in E(D_0), \tag{6j}$$

$$f_{ij}^{st} \leq F_{ij}^s \quad (i,j) \in A, \{s,t\} \in E(D_0), \tag{6k}$$

$$f_{ij}^{st} \leq F_{ij}^t \quad (i,j) \in A, \{s,t\} \in E(D_0), \tag{6l}$$

$$F_{ij}^s + F_{ij}^t - f_{ij}^{st} \leq g_{ij} \quad (i,j) \in A, \{s,t\} \in E(D_0), \tag{6m}$$

$$\sum_{k\in W_{ij}} \left(F_{ik}^t + F_{ki}^s - f_{ik}^{st} - f_{ki}^{st}\right) \leq \sum_{k\in W_{ij}} \pi_{ik}^s \quad (i,j) \in A, \{s,t\} \in E(D_0), \tag{6n}$$

$$\sum_{k\in W_{ij}} F_{ik}^t \leq \sum_{k\in W_{ij}} \pi_{ik}^0 \quad (i,j) \in A, t \in D_0, \tag{6o}$$

$$f \in \{0,1\}^{A\times E(D_0)}. \tag{6p}$$

By (6j), it is ensured that the joint flow is non-increasing by increasing distance from the root s_0. While constraints (6k)–(6l) ensure $f_{ij}^{st} = 0$ if $F_{ij}^s = 0$ or $F_{ij}^t = 0$, inequalities (6m) impose $f_{ij}^{st} = 1$ if $F_{ij}^s = F_{ij}^t = 1$. By virtue of (6l), constraints (6m) replace the weaker capacity constraints (2a). Finally, (6n)–(6o) are valid inequalities justified by:

Proposition 5 *Assume (F, f, g, π) satisfies (2b)–(2l), (1k), (6j)–(6m), and (6p). Let $T^0 \subseteq G_g$ be the corresponding arborescence spanning D (see Lemma 2), and let T^s be the arborescence obtained by redirecting the arcs in T^0 such that they point away from $s \in D$. Then, for all nodes i and j $(i \neq j)$ spanned by T^0,*

$$F_{ij}^t + F_{ji}^s - f_{ij}^{st} - f_{ji}^{st} = \begin{cases} 1, & \textit{if } (i,j) \textit{ is an arc on the path from } s \textit{ to } t \textit{ in } T^s, \\ 0, & \textit{otherwise.} \end{cases}$$

Proof Because $g_{ij} + g_{ji} \leq 1$, we have $F_{ij}^t + F_{ji}^s \leq 1$. Constraints (6k)–(6l) thus yield $f_{ij}^{st} + f_{ji}^{st} \leq 1$, and $F_{ij}^t + F_{ji}^s - f_{ij}^{st} - f_{ji}^{st} \in \{0,1\}$ follows. Inequalities (6m) in combination with (6k)–(6l) also imply $f_{ij}^{st} = F_{ij}^t F_{ji}^s$.

The path from s to t in T^s contains (i,j) iff $(i,j) \in A_{F^t} \setminus A_{F^s}$ or $(j,i) \in A_{F^s} \setminus A_{F^t}$. This condition is equivalent to $F_{ij}^t = 1 - F_{ij}^s = 1$ or $F_{ji}^s = 1 - F_{ji}^t = 1$, which holds iff $F_{ij}^t(1 - F_{ij}^s) + F_{ji}^s(1 - F_{ji}^t) = F_{ij}^t - f_{ij}^{st} + F_{ji}^s - f_{ji}^{st} = 1$. □

Constraints (6n) reflect the observation that if (i,j) is on the path from s to t in T^s, then the cost of the most expensive arc leaving i in T^s is at least

p_{ij}. Validity of (6o) is explained analogously, as $F_{ij}^t = 1$ iff (i, j) is on the path from s_0 to t in T^0.

6.3 Constraint Generation Scheme

Owing to the large number of constraints in the stronger models, LP($\mathcal{X}_3$) and LP($\mathcal{F}_3$) are impractical for computing lower bounds, even in fairly small instances. To enable computation of strong lower bounds, we first solve a reduced version of the LP, where some of the constraints are omitted. Some of the relaxed constraints that are violated in the solution are next added, and the process is repeated until no violations exist. This approach is known as a *constraint generation* (CG) scheme.

6.3.1 Implementation

Preliminary computational experiments indicate that solving LP($\mathcal{F}_k$) ($k = 1, 2$) in order to compute a stronger bound than the one output from LP($\mathcal{X}_k$) is also more time-consuming. The disproportion in running time is even more considerable in a comparison between LP($\mathcal{F}_3$) and LP($\mathcal{X}_3$) (see Section 8.2). For these practical reasons, we apply the CG idea to LP($\mathcal{X}_3$).

Consider an optimal solution (X, y, π) to LP($\mathcal{X}_2$), and let X and π be fixed. Whether there exists an $x \in [0, 1]^{A \times A(D)}$ satisfying (6d)-(6f) can be checked by solving a maximum flow problem for each $(s, t) \in A(D)$. The constraints state that it should be feasible to send one unit of flow from source $s \in D$ to destination $t \in D_s$, such that the flow on arc (i, j) is no more than X_{ij}^s. Further, the symmetry constraints (6g) implicitly impose the upper bound X_{ji}^t on the same flow. Finally, the side constraints (6h) impose upper bounds on the total flow on subsets of arcs leaving a given node. By applying the symmetry $x_{ij}^{st} = x_{ji}^{ts}$ also to (6h), the extended maximum flow model, denoted $\mathcal{MF}$, is completed as (the x-variables do not have the superscripts s and t as $\{s, t\}$ is fixed within each model instance):

$$\max \sum_{i \in V_s} x_{si}, \tag{9a}$$

s.t.

$$\sum_{j \in V_i} (x_{ij} - x_{ji}) = 0 \qquad i \in V \setminus \{s,t\}, \tag{9b}$$

$$x_{ij} \leq \min\{X^s_{ij}, X^t_{ji}\} \qquad (i,j) \in A, \tag{9c}$$

$$\sum_{k \in W_{ij}} x_{ik} \leq \sum_{k \in W_{ij}} \pi^s_{ik} \qquad (i,j) \in A, \tag{9d}$$

$$\sum_{k \in W_{ij}} x_{ki} \leq \sum_{k \in W_{ij}} \pi^t_{ik} \qquad (i,j) \in A, \tag{9e}$$

$$x \in [0,1]^A. \tag{9f}$$

For $Q \subseteq E(D) = \{\{s,t\} \in E : s,t \in D\}$, let $\mathcal{X}_2 + Q$ denote the extension of $\mathcal{X}_2$ by constraints (6d)–(6h) for all $\{s,t\} \in Q$. In the CG procedure described in Algorithm 1, we first solve LP($\mathcal{X}_2$) to obtain solution vectors X and π. For each $\{s,t\} \in E(D)$, we then check whether the relaxed constraints (6d)–(6h) can be satisfied, given the values of X and π. This question is answered by solving $\mathcal{MF}$ since the objective function in $\mathcal{MF}$ takes optimal value 1 iff the constraints are satisfiable. In each iteration, relaxed constraints corresponding to a selection of pairs $\{s,t\}$ for which the maximum flow is below 1, are added to the original model. How to make this selection is discussed next.

```
Q ← ∅
do
    (X, y, π) ← optimal solution to LP(X_2 + Q)
    for {s,t} ∈ E(D) \ Q do
        v_st ← optimal objective function value in MF
    end
    Q' ← {{s,t} ∈ E(D) \ Q : v_st < 1}
    Select some ΔQ ⊆ Q' such that ΔQ ≠ ∅ if Q' ≠ ∅
    Q ← Q ∪ ΔQ
while ΔQ ≠ ∅;
```

Algorithm 1: Constraint generation

6.3.2 Constraint selection

Preliminary experiments have demonstrated that in many instances, the optimal values of X and π in LP($\mathcal{X}_2$) are too restrictive to allow for unit maximum flow. For most $\{s,t\} \in E(D)$, the objective function of $\mathcal{MF}$ takes a value below 1 in optimum. Adding the constraints corresponding to all $\{s,t\}$ for which this occurs thus involves computations almost as extensive as solving LP($\mathcal{X}_3$) directly.

Selection of pairs $\{s,t\}$ to be included in ΔQ in Algorithm 1 is based on the following intuitive considerations:

Adjacency matters: Extending Q by $\{s,t\}$ is more likely to imply $v_{uw}=1$ in the following iteration if $\{s,t\} \cap \{u,w\} \neq \emptyset$ ($\{u,w\} \in E(D) \setminus Q$).

Extent of violation matters: The growth of $z\,(\mathrm{LP}(\mathcal{X}_2 + Q))$ increases by decreasing value of v_{st}.

Balancing the needs for rapid growth in $z\,(\mathrm{LP}(\mathcal{X}_2 + Q))$ and small value of $|Q|$, we pursue the CG scheme as follows: In the graph (D, Q'), where $Q' = \{\{s,t\} \in E(D) : v_{st} < 1\}$, let the edge weights be $1 - v_{st}$. Let ΔQ be a matching of maximum weight in (D, Q'). Thus, for each $\{u,w\} \in E(D) \setminus Q$ where constraints (6d)–(6h) are not satisfiable given the current values of X and π, Q is extended by an adjacent edge.

7 Computing near-optimal solutions

Instances of size that prohibits computation of the optimal solution should be approached by some fast, but possibly inexact method. Imposing a time bound on an IP-solver, and applying it to any of the models introduced in Section 4, readily gives such a method. Alternatively, a heuristic method dedicated to SMT can be developed.

To the best of our knowledge, the scientific literature offers few heuristic methods for SMT besides those studied by Ivanova (2016). However, recent progress by Pajor et al. (2018) on heuristic methods for the related minimum Steiner tree problem can smoothly be exploited in order to generate analogous methods for SMT. The main algorithmic components in the most basic heuristic are a construction algorithm, a local improvement algorithm, and an algorithm for merging two Steiner trees.

A set $\mathcal{T} = \left(T_1, \ldots, T_{|\mathcal{T}|}\right)$ of at most γ distinct Steiner trees, ordered by non-decreasing costs, is maintained throughout the heuristic (Pajor et al. 2018). Construction and local improvement are applied in order to generate a new tree T of presumably low cost. Diversity in $\mathcal{T}$ is achieved by perturbing the edge costs randomly before construction and local improvement are applied. If $|\mathcal{T}| < \gamma$, T is included in $\mathcal{T}$. If $|\mathcal{T}| = \gamma$, and the cost of T is below the cost of T_γ, T_γ is replaced by T in $\mathcal{T}$. Another Steiner tree, $\tilde{T}$, is generated by merging T with a randomly chosen $T_r \in \mathcal{T}$ ($r = 1, \ldots, \gamma$), and $\tilde{T}$ replaces T_γ in $\mathcal{T}$ if its cost is smaller.

Taking advantage of the recently published construction and local improvement algorithms for SMT (Ivanova 2016), Alg. 2 shows how the method by Pajor et al. (2018) is adapted to our problem. The procedure $\mathit{SMTHeur}(G, D, p)$ produces a multicast tree T by first applying the construction algorithm to the input (G, D, p), and next applying local improvement to the constructed tree. Following Pajor et al. (2018), the perturbed cost $\hat{p}_{ij}$ of edge $\{i, j\}$ is put equal

```
while the time assigned to the heuristic is not expired do
    for {i,j} ∈ E do p̂_ij ← random number uniformly distributed on [½p_ij, 3/2 p_ij]
    T ← SMTHeur(G, D, p̂)
    if |𝒯| < γ then 𝒯 ← 𝒯 ∪ {T}
    else
        if c(T) < c(T_γ) then 𝒯 ← 𝒯 \ {T_γ}, 𝒯 ← 𝒯 ∪ {T}
        r ← random number uniformly distributed on {1, ..., γ}
        for {i,j} ∈ E_T ∩ E_{T_r} do p̃_ij ← p_ij
        for {i,j} ∈ (E_T ∪ E_{T_r}) \ (E_T ∩ E_{T_r}) do
            ν ← random number uniformly distributed on {100, ..., 500}
            p̃_ij ← ν p_ij
        end
        for {i,j} ∈ E \ E_T \ E_{T_r} do p̃_ij ← 1000 p_ij
        T̃ ← SMTHeur(G, D, p̃)
        if c(T̃) < c(T_γ) then 𝒯 ← 𝒯 \ {T_γ}, 𝒯 ← 𝒯 ∪ {T̃}
    end
end
return T_1
```

Algorithm 2: Outline of the heuristic method

to a random number with expected value p_{ij}. Further, in the merge process, adjusted weights $\tilde{p}_{ij}$ are determined in order to promote edges that appear in both T and the randomly selected tree $T_r \in \mathcal{T}$. The cost of edges appearing in exactly one of the trees are thus magnified. For the purpose of diversification, the factor by which the cost is multiplied is chosen randomly. To discourage the appearance of edges outside both T and T_r in the merge $\tilde{T}$, their cost is magnified by a factor dominating the factors drawn for edges that occur in one of the trees.

A few minor simplification over the approach taken by Pajor et al. (2018) are made in Alg. 2: Rather than selecting the tree to leave $\mathcal{T}$ randomly, we consistently remove T_γ when a new tree is added. Also, for each new tree T constructed, Alg. 2 makes only one attempt to merge it with some $T_r \in \mathcal{T}$. This contrasts the method in the cited work, where $\tilde{T}$ is chosen as the best tree found in a sequence of merges.

8 Experimental Evaluation

The practical part of this work focuses on comparison of models presented in Sections 4 and 6. Instances of given number of nodes and destinations are generated with random coordinates uniformly distributed on the square with corners at [0, 0] and [100, 100]. All edge costs equal the square of the distance between the end nodes. A time limit of 20 CPU-minutes is imposed on each run, unless otherwise is stated. All experiments are run on an Intel Core 2 Quad CPU at 2.83 GHz and 8 GB RAM. The models are implemented in Java with Concert technology and solved using Cplex 12.5.1 optimizer.

8.1 Comparing the computational burden and the lower bound of the LP relaxations

The first experiments give an overview of the LP relaxations of the models with respect to their strength and computational time. Tab. 1 summarizes optimal objective function values of all LP relaxations. Each table entry is the value of $z(\mathrm{LP}(\mathcal{M}))/z^*$ for the model $\mathcal{M}$ corresponding to the column, averaged over 25 instances of the size corresponding to the row. Here, z^* denotes the minimum power. Entries labeled by an asterisk correspond to runs that, in at least one of the 25 instances, are interrupted because the time bound was reached. The dual simplex method is used to solve the LPs, such that a lower bound on $z(\mathrm{LP}(\mathcal{M}))$ is available upon interruption. The rows are clustered by $|V|/|D|$ ratio.

Confirming Props. 3 and 4, model $\mathrm{LP}(\mathcal{F}_k), (k = 1, 2)$ yields a consistently tighter lower bound than does $\mathrm{LP}(\mathcal{X}_k)$. The difference is more prominent for $k = 1$, where the bound obtained by $\mathrm{LP}(\mathcal{F}_1)$ is 13% tighter in average, while $\mathrm{LP}(\mathcal{F}_2)$ gives a 5% tighter bound in average. A comparison between $\mathrm{LP}(\mathcal{X}_3)$ and $\mathrm{LP}(\mathcal{F}_3)$ shows that they give identical bounds in all instances that are solved to optimality within the time limit by use of $\mathrm{LP}(\mathcal{F}_3)$. Using $\mathrm{LP}(\mathcal{F}_3)$, this is possible in all instances. In the instances in question, the bounds provided by the stronger models are significantly tighter than those of $\mathrm{LP}(\mathcal{F}_2)$, and close or equal to the integer optimum.

$\|V\|$	$\|D\|$	$\mathrm{LP}(\mathcal{X}_1)$	$\mathrm{LP}(\mathcal{F}_1)$	$\mathrm{LP}(\mathcal{X}_2)$	$\mathrm{LP}(\mathcal{F}_2)$	$\mathrm{LP}(\mathcal{X}_3)$	$\mathrm{LP}(\mathcal{F}_3)$
12	8	78.37	82.69	85.40	86.95	99.92	99.92
15	10	78.19	82.35	85.04	86.89	99.88	99.88
18	12	74.05	79.78	80.65	83.47	94.39	55.27*
14	7	74.83	80.65	83.13	84.86	99.92	99.92
16	8	72.25	80.39	82.20	85.97	99.78	99.78
18	9	65.68	77.17	74.48	79.84	99.70	74.17*
15	5	65.99	78.08	80.87	86.11	100.00	100.00
18	6	66.08	77.13	79.71	84.23	99.94	99.94
21	7	62.26	76.05	75.89	83.33	99.96	99.27*

Table 1: Average LP bounds as percentages of integer optimum

Average running times are reported in Tab. 2. The 20 CPU-minutes time restriction manifests itself when applying the strongest LP relaxations to some instances. Solving $\mathrm{LP}(\mathcal{F}_k), (k = 1, 2, 3)$ is consistently more time-consuming than solving $\mathrm{LP}(\mathcal{X}_k)$, which demonstrates a trade-off between strength and running time. Moreover, the running times of $\mathrm{LP}(\mathcal{X}_3)$ and $\mathrm{LP}(\mathcal{F}_3)$ are, in instances of the actual size, considerably longer than the time it takes to solve $\mathcal{F}_1$ to optimality (see the last column) using branch-and-bound (B&B). This implies that both LP($\mathcal{X}_3$) and LP($\mathcal{F}_3$) are impractical in their basic form. Bounds corresponding to the other models are computed in a few seconds.

$\|V\|$	$\|D\|$	LP($\mathcal{X}_1$)	LP($\mathcal{F}_1$)	LP($\mathcal{X}_2$)	LP($\mathcal{F}_2$)	LP($\mathcal{X}_3$)	LP($\mathcal{F}_3$)	$\mathcal{F}_1$
12	8	0	0	0	0	8	26	2
15	10	1	1	1	1	146	566	16
18	12	3	4	3	5	1161	1171*	115
14	7	0	0	0	0	5	46	5
16	8	1	1	1	1	72	304	18
18	9	1	2	2	2	326	1140*	65
15	5	0	0	0	0	1	8	3
18	6	0	0	0	0	9	109	12
21	7	1	2	2	3	107	884*	67

Table 2: Comparison of average running time in CPU-seconds

8.2 Computing optimal solutions

Small instances are often solved to optimality relatively quickly with B&B. We investigate what is the maximum size of instances that can be solved to optimality within 20 CPU-minutes using models $\mathcal{X}_1$ and $\mathcal{F}_1$. The experiments compare sets of instances with different values of $|V|/|D|$. For instances of the size given in columns 1–2, Table 3 shows the average running time in CPU-seconds (columns 3–4), the number of instances solved to optimality (columns 5–6), and the average ratio between the upper and lower bounds on the minimum cost (columns 7–8) upon interruption of the solver. Each table value is obtained after solving 25 instances of given size.

Unlike the case of LP relaxation, the running time needed to solve $\mathcal{F}_1$ by B&B is shorter than the time needed to solve $\mathcal{X}_1$. This is observed for all instance sizes. As a consequence, the number of instances solved to optimality within the given time limit is smaller for $\mathcal{X}_1$ than for $\mathcal{F}_1$. B&B applied to $\mathcal{F}_1$ solves to optimality all the instances with no more than 20 nodes, as well as those where $|V| = 21$ and $|D| = 7$. When the weaker model $\mathcal{X}_1$ is used, more time is needed in some of these instances. Furthermore, when instances are not solved to optimality, $\mathcal{F}_1$ leaves a tighter optimality gap than does $\mathcal{X}_1$. Note that the lower the value of $|V|/|D|$, the more difficult are the instances to solve. This applies to both $\mathcal{F}_1$ and $\mathcal{X}_1$.

8.3 Comparison with alternative methods

We now investigate the ability of IP models $\mathcal{X}_1$ and $\mathcal{F}_1$ to produce good feasible solutions when a time limit is imposed. We compare the quality of obtained

		average time [s]		# solved		average remaining gap[%]	
$\|V\|$	$\|D\|$	$\mathcal{X}_1$	$\mathcal{F}_1$	$\mathcal{X}_1$	$\mathcal{F}_1$	$\mathcal{X}_1$	$\mathcal{F}_1$
18	12	139	84	24	25	0.52	0.00
21	14	1006	789	10	18	9.64	5.08
24	16	1161	1137	3	4	21.04	20.52
20	10	387	190	21	25	3.00	0.00
22	11	880	637	14	22	7.16	0.64
24	12	1169	997	3	10	16.44	10.44
21	7	282	113	21	25	2.56	0.00
24	8	650	442	16	21	8.64	2.72
27	9	1146	1013	4	8	22.80	14.64

Table 3: Results obtained from B&B applied to $\mathcal{X}_1$ and $\mathcal{F}_1$

		5 min			10 min			20 min		
$\|V\|$	$\|D\|$	$\mathcal{X}_1$	$\mathcal{F}_1$	Alg. 2	$\mathcal{X}_1$	$\mathcal{F}_1$	Alg. 2	$\mathcal{X}_1$	$\mathcal{F}_1$	Alg. 2
18	12	100.1	100.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
21	14	107.9	108.8	100.0	102.7	103.2	100.0	101.2	100.7	100.0
24	16	153.4	115.6	100.0	126.2	114.2	100.0	104.3	112.7	100.0
20	10	100.6	100.1	100.0	100.1	100.0	100.0	100.1	100.0	100.0
22	11	102.8	103.9	100.0	101.3	100.7	100.1	100.7	100.2	100.1
24	12	108.2	113.4	100.0	105.0	105.5	100.0	103.1	106.2	100.0
21	7	100.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
24	8	102.8	101.9	100.0	102.1	100.7	100.0	101.1	100.1	100.0
27	9	133.2	118.7	100.0	111.3	109.1	100.0	108.2	104.1	100.0

Table 4: Comparison between the metaheuristic algorithm (Alg. 2) and interrupted B&B applied to IP models $\mathcal{X}_1$ and $\mathcal{F}_1$

solutions with those calculated by Alg. 2. The results for time restrictions 5, 10 and 20 CPU-minutes, respectively, are reported in Tab. 4. These experiments are carried out on the instance sets that are investigated in the experiments reported in Section 8.2.

For a given instance, we define the relative performance of a method as the objective function value obtained by the method relative to the best performing method in that instance. For each time bound imposed, and for each instance set, we report the relative performances of $\mathcal{X}_1$, $\mathcal{F}_1$, and Alg. 2, averaged over all instances in the set. For example, if all methods find an optimum in every instance, the three corresponding values are all 100.0%. Thus, the lowest value in a given instance set indicates the best performing method.

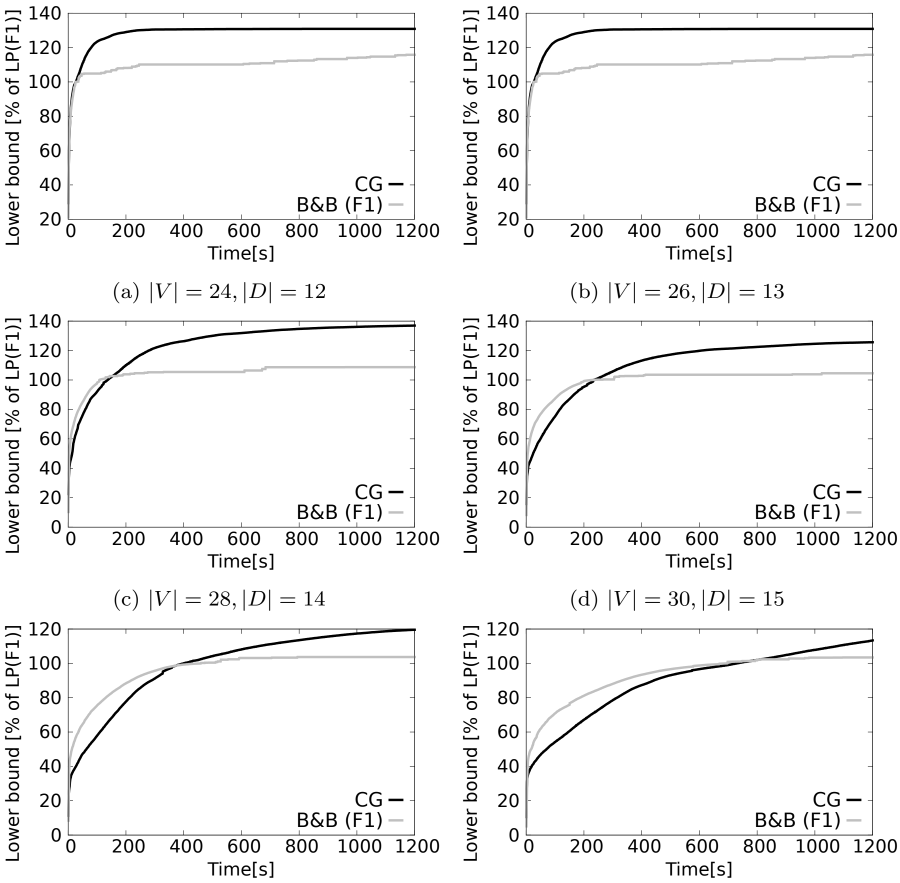
Table 4 reveals that Alg. 2 gives the best average results for all combinations of time limit and instance set. Even when the time bound is set beyond what is needed to compute an optimal solution by submitting or $\mathcal{F}_1$ to B&B (see Tab. 3), Alg. 2 produces no worse solutions, and is thus able to identify an optimal solution in each considered instance. Although Alg. 2 fails to prove optimality in general, it appears to be a better procedure than B&B for computing near-optimal solutions within a given time bound.

8.4 Experiments on strong lower bounds

As the instance size increases, exact solution of the SMT problem becomes unrealistic. To assess the quality of fast and possible inexact methods, such as Alg. 2, or the construction methods by Ivanova (2016), access to tightest possible lower bounds is crucial. In the next set of experiments, Section 8.1 indicates that $z\,(\mathrm{LP}(\mathcal{X}_3))$ is a strong bound, but solving $\mathrm{LP}(\mathcal{X}_3)$ requires excessive computations. At this point, the CG scheme (Alg. 1) comes into play. Because $\mathrm{LP}(\mathcal{F}_1)$ is solved quickly, lower bounds can alternatively be computed by applying B&B to $\mathcal{F}_1$ for a restricted amount of time. While the initial bound $z\,(\mathrm{LP}(\mathcal{F}_1))$ might be weak, fast solution of $\mathrm{LP}(\mathcal{F}_1)$ enables rapid growth in the B&B-tree, and, possibly, also in the corresponding lower bound.

We compare the progress of the lower bounds obtained by the CG method with the one obtained by B&B applied to $\mathcal{F}_1$ as a function of running time. Instances with $|V| = 2|D| \in \{24, 26, \dots, 34\}$ are investigated, with 5 instances for each value of $|V|$. The results are illustrated in Fig. 3. Each curve represents the progress of the lower bound within each instance set. The horizontal axes represent running times ranging from 0 to 20 CPU-minutes. The vertical axes correspond to the lower bounds as a percentage of $z\,(\mathrm{LP}(\mathcal{F}_1))$, averaged over all instances in the set. The tightness of the bounds increases with time, and it can be observed from the graphs that the CG bound becomes tighter at some point. We express the lower bounds relative to $z\,(\mathrm{LP}(\mathcal{F}_1))$, as indicated by the vertical axes. The lower bound obtained from B&B grows sharply while the dual simplex method is solving the root problem of the B&B tree, which goes on until the curve reaches 100%. After that point, the bound grows rather slowly. In contrast, CG produces a bound that grows faster, even after it has reached the value $z\,(\mathrm{LP}(\mathcal{X}_1))$.

Reflecting the progress of the dual simplex method applied to $\mathrm{LP}(\mathcal{X}_2)$ and $\mathrm{LP}(\mathcal{F}_1)$, respectively, both CG and B&B increase the lower bound sharply at the beginning. While solution of $\mathrm{LP}(\mathcal{F}_1)$ is still in progress, the growth of the lower bound is faster than what is observed for CG (solution of $\mathrm{LP}(\mathcal{F}_1)$). This is particularly apparent for instances of size 28 and beyond (Fig. 3c - 3f). Once $\mathrm{LP}(\mathcal{F}_1)$ is solved, the increase slows down substantially, and remains modest until the time limit is reached. An analogous, but more moderate, gradual slowdown is also observed for CG. In each instance set, the two curves intersect at some point in time. From this moment on, CG provides a tighter lower bound. Experiments reported in Fig. 3 thus indicate that CG is the more suitable method for obtaining tight lower bounds within a given time.



(a) $|V| = 24, |D| = 12$ (b) $|V| = 26, |D| = 13$

(c) $|V| = 28, |D| = 14$ (d) $|V| = 30, |D| = 15$

(e) $|V| = 32, |D| = 16$ (f) $|V| = 34, |D| = 17$

Fig. 3: Comparison of lower bounds obtained by CG and B&B on model $\mathcal{F}_1$. Each figure depicts the progress of lower bounds for instances with varying number of destination and non-destination nodes. The lower bounds are obtained by applying the CG scheme to model $\mathcal{F}_1$ (black curves), and by running the B&B algorithm (grey curves), respectively.

9 Conclusion and Future Work

We have presented and analyzed two approaches of modelling the SMT problem as an integer linear program. The first formulation, $\mathcal{X}_1$, is based on a formulation of the broadcast version of the problem, while the second one, $\mathcal{F}_1$, is an extension of network flow formulation of the minimum Steiner tree problem. Both formulations are improved by valid inequalities resulting in models $\mathcal{X}_2$ and $\mathcal{F}_2$, respectively. Further strengthening is achieved by introducing variables of higher dimension and associated constraints, which leads to models $\mathcal{X}_3$ and $\mathcal{F}_3$. A theoretical analysis of the formulations shows that $\mathcal{F}_1$ and $\mathcal{F}_2$ are at least as strong as $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively. It is conjectured that an analogous relation holds between $\mathcal{F}_3$ and $\mathcal{X}_3$. Experimental evaluation reveals that instances with up to approximately 20 nodes are practically solvable to optimality. Subsequent investigation suggests that the LP relaxations of $\mathcal{F}_3$ and $\mathcal{X}_3$ are fairly strong, but the running time prohibits direct solution. Nevertheless, experiments also show that a constraint generation procedure applied to $\mathcal{X}_3$, developed in the current work, provides tight lower bounds. The procedure also demonstrates that when the number of nodes is no more than 30, many instances of the LP relaxation of $\mathcal{X}_3$ have an integer optimum.

Follow-up research should uncover whether the conjecture on the relation between the LP relaxations of $\mathcal{F}_3$ and $\mathcal{X}_3$ holds. A search for facet-defining valid inequalities, or even convex hull formulations, is an interesting and ambitious direction of future work. Owing to the limited size of solvable instances, there is a potential in studying inexact methods, such as approximation algorithms and problem specific heuristics. In particular, experiments on a simple metaheuristic documented in the current work, suggest that fast and inexact methods have a potential to compute solutions close to optimality in large instances. Instance preprocessing combined with variable fixing can possibly increase the size of instances solvable to optimality with the current models.

References

Althaus E, Călinescu G, Măndoiu II, Prasad S, Tchervenski N, Zelikovsky A (2003) Power efficient range assignment in ad-hoc wireless networks. In: Proceedings IEEE WCNC03, pp 1889–1894

Altinkemer K, Salman FS, Bellur P (2005) Solving the minimum energy broadcasting problem in ad hoc wireless networks by integer programming. In: Proceedings INOC 2005, pp B2.635–B2.642

Bauer J, Haugland D, Yuan D (2008) Analysis and computational study of several integer programming formulations for minimum-energy multicasting in wireless ad hoc networks. Networks 52(2):57–68

Bein D, Zheng SQ (2010) Energy efficient all-to-all broadcast in all-wireless networks. Information Sciences 180(10):1781–1792

Bhukya WN, Singh A (2014) *p*-shrink: A Heuristic for Improving Minimum All-to-All Power Broadcast Trees in Wireless Networks. In: Proceedings of Ninth International Conference on Wireless Communication and Sensor Networks 2014 Vol. 299, pp 61–69

Čagalj M, Hubaux J-P, Enz C (2002) Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In: Proceedings ACM MOBICOM 2002 pp 172–182

Clementi A, Penna P, Silvestri R (1999) Hardness results for the power range assignment problem in packet radio networks. In: Hochbaum D et al. (eds) RANDOM-APPROX99, LNCS 1671:197–208

Das AK, Marks RJ, El-Sharkawi M, Arabshani P, Gray A (2003) Minimum power broadcast trees for wireless networks: integer programming formulations. In: Proceedings IEEE INFOCOM 2003 Vol. 2, pp 1001–1010

Das AK, Marks RJ, El-Sharkawi E, Arabshahi P, Gray A (2005) Optimization methods for minimum power bidirectional topology construction in wireless networks with sectored antennas. In: Proceedings IEEE GLOBECOM 05, pp 3962–3968

Goemans MX, Myung Y-S (1993) A catalog of Steiner tree formulations. Networks 23(1):19–28

Guo S, Yang O (2004) Minimum Energy Multicast Routing for Wireless Ad-hoc Networks with Adaptive Antennas In: Proceedings of the 12th IEEE International Conference on Network Protocols ICNP 2004, pp 151–160

Halgamuge MN, Zukerman M, Ramamohanarao K (2009) An estimation of sensor energy consumption. Progress In Electromagnetics Research B 12:259–295

Haugland D, Yuan D (2011) Compact integer programming models for power-optimal trees in ad hoc wireless networks. In: Kennington J, Olinick E, Rajan D (eds) Wireless network design - optimization models and solution procedure. International series in operations research & management science. Springer, New York, pp 219–246

Hsiao P-C, Chiang T-C, Fu L-C (2013) Static and dynamic minimum energy broadcast problem in wireless ad-hoc networks: A PSO-based approach and analysis Applied Soft Computing 13:786–4801

Ivanova M (2016) Shared Multicast Trees in Ad Hoc Wireless Networks. In: Cerulli R, Fujishige S, Mahjoub A (eds) Combinatorial Optimization ISCO 2016, LNCS 9849:273–284

Kirousis LM, Kranakis E, Krizanc D, Pelc A (1997) Power consumption in packet radio networks. In: Reischuk M (ed): STACS'97 Proceedings, LNCS 1200:363–374

Leggieri V, Nobili P, Triki C (2008) Minimum power multicasting problem in wireless networks. Math Meth Oper Res 68:295–311

Montemanni R, Gambardella LM (2004) Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. Comput Oper Res 31(10):1667–1680

Montemanni R, Leggieri V (2011) A branch and price algorithm for the minimum power multicasting problem in wireless sensor networks. Math Meth Oper Res 74:327–342

Pajor T, Uchoa E, Werneck RF (2018) A robust and scalable algorithm for the Steiner problem in graphs. Math Prog Comp 10:69–118

Papadimitriou I, Georgiadis L (2006) Minimum-energy Broadcasting in Multi-hop Wireless Networks Using a Single Broadcast Tree. Mobile Netw Appl 11(3):361–375

Polzin T, Daneshmand SV (2001) A comparison of Steiner tree relaxations. Discrete Appl Math 112(1–3):241–261

Wieselthier JE, Nguyen GD, Ephremides A (2000) On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks, In: Proceedings IEEE INFOCOM 2000 Vol. 2, pp 585–594

Yuan D (2005) An integer programming approach for the minimum-energy broadcast problem in wireless networks. In: Proceedings INOC 2005, pp B2.643–B2.650

Yuan D, Bauer J, Haugland D (2008) Minimum-energy broadcast and multicast in wireless networks: An integer programming approach and improved heuristic algorithms. Ad Hoc Netw 6(5):696–717

Yuan D, Haugland D (2012) Dual Decomposition for Computational Optimization of Minimum-Power Shared Broadcast Tree in Wireless Networks. IEEE T Mobile Comput 12(11):2008–2019