Stronger Path-based Extended Formulation for the Steiner Tree Problem*

Bartosz Filipecki[†], Mathieu Van Vyve[‡]

May 22, 2019

Abstract

The Steiner tree problem (STP) is a classical NP-hard combinatorial optimization problem with applications in computational biology and network wiring. The objective of this problem is to find a minimum cost subgraph of a given undirected graph G with edge costs, that spans a subset of vertices called terminals. We present currently used linear programming formulations of the problem based on two different approaches - the bidirected cut relaxation (BCR) and the hypergraphic formulations (HYP), the former offering better computational performance, and the latter better bounds on the integrality gap. As our contribution, we propose a new hierarchy of path-based extended formulations for STP. We show that this hierarchy provides better integrality gaps on graph instances used to define the worst-case lower bounds on the integrality gap for both BCR and HYP. We prove that each consecutive level of our hierarchy is at least as strong as the previous one. Additionally, we also present numerical results showing that several difficult STP instances can be solved to integer optimality by using this hierarchy. Our approach can be adapted to variants of STP or applied to hypergraphic formulations for further potential improvement on the integrality gap bounds, in exchange for additional computational effort.

Keywords: Steiner Tree, Extended formulation, Integrality gap, Linear programming relaxation, Mixed-integer linear programming, Flow-based formulation

*This research was supported by the Marie Curie ITN 'MINO' and the P7/36 PAI project COMEX.

†Corresponding author. Formerly: CORE, Université Catholique de Louvain, Belgium. Currently: Technische Universität Chemnitz, Germany.

‡CORE & Louvain School of Management, Université Catholique de Louvain, Belgium.

1 Introduction

The Steiner Tree Problem (STP) is an NP-hard problem [20] with applications ranging from computational biology to network routing and design [19]. Given an undirected graph $G = (V, E)$ with associated edge-cost function $c : E \to \mathbb{R}^+$ and a set of terminals $R \subseteq V$, the objective is to find a minimum-cost subgraph $T \subseteq G$ that spans R. The vertices of G that belong to $V \setminus R$ are called Steiner nodes. As existence of a cycle would imply that the cost could be decreased, the optimal solution is thus a tree and hence the name of the problem.

This paper focuses on strengthening linear programming (LP) relaxations of STP. Current algorithms usually depend on one of two LP relaxations - the bidirected cut relaxation (BCR) [5, 7, 26, 29] and the Hypergraphic relaxation (HYP) [3, 8, 28]. Their corresponding formulations gave rise to numerous optimization algorithms. Most recently, a HYP-based 1.39-approximation algorithm was proposed by Byrka et al. [2], with a stronger analysis carried out later on by Goemans et al. [15]. The latter reference also proved that the integrality gap for HYP is at most 1.39. Until then, the leading HYP-based algorithm was the 1.55-approximation one proposed by Robins and Zelikovsky [27], which was not LP-based. For BCR, which is more efficiently solvable than HYP, the best-known, but somewhat trivial, upper bound on the integrality gap is 2 [26]. It results by taking a metric closure of graph G and obtaining a spanning tree for the terminals. For lower bounds on the integrality gap, Chlebík and Chlebíková [4] proved that STP is NP-hard to approximate within ratio $96/95 \approx 1.01$. On several instances used to prove the worst-case bounds, HYP generally attains smaller integrality gaps than BCR. For a graph known as Skutella's graph the HYP and BCR formulations both attain an $8/7 \approx 1.14$ integrality gap [23]. For BCR, there is an entire family of graphs suggested by Goemans [13], for which the same gap is also attained. Based on Skutella's graph, an additional family of graphs is obtained with a $36/31 \approx 1.16$ integrality gap for BCR [2]. Finally, it has been proven by Feldmann et al. [9] that BCR and HYP are polyhedrally equivalent (i.e., they yield the same LP bound) for graphs that do not contain a Steiner vertex with three neighboring Steiner vertices. However, solving HYP is computationally expensive and therefore BCR-based relaxations are normally preferred in the design of exact algorithms. On the practical side, the 11th DIMACS challenge (2014) has inspired several state–of-the-art algorithms for STP in recent years. This includes work by Hougardy et al. [18], Fu et al. [11], Fischetti et al. [10] and Gamrath et al. [12] to name a few.

Our main contribution is to strengthen BCR and HYP formulations by introducing additional variables to a standard multicommodity flow formulation of STP. The new variables represent the flow of commodities along paths of G and not, simply, flows on corresponding single edges. We show that our hierarchy of extended formulations improves the lower bound on the integrality gap in all specific graph instances previously used to establish worst cases on LP relaxation gaps. This is of particular interest to BCR, the formulation favored for designing exact algorithms. Indeed, it might be necessary to strengthen BCR if its upper bound of 2 is to be improved. To our knowledge, ours is the second flow-based hierarchy of extended formulations proposed for the Steiner tree problem. The first one is due to Polzin and Daneshmand [6,24]. Both of these hierarchies improve the standard BCR formulation. Finally, we present numerical results showing how our hierarchy decreases or completely closes the integrality gap on several difficult STP instances.

We simultaneously consider previously defined undirected graphs $G = (V, E)$ and their corresponding bidirected graphs $D = (V, A)$ with the set of arcs $A = \{(i, j) | \{i, j\} \in E\}$. We use the standard notation $\{i, j\} \in E$ for an edge of G between vertices i and j. Likewise, $(i, j) \in A$ is used for an arc of D that points from i to j. Indices ij will be used to denote either edges of G or arcs of D. We also use the classical notation $x^+ = \max\{0, x\}$ and $\delta^+(C) = \{(i, j) \in A | i \in C,\ j \notin C\}$. This means that $\delta^+(C)$ is the set of arcs of D that point from a vertex in C to a vertex in $V \setminus C$. Finally, for conciseness we define $\mathcal{RK} = \{(r, k) : r \in R, k \in R \setminus \{r\}\}$ as the set of pairs of distinct terminals.

1.1 LP relaxations for STP

We now introduce the standard LP relaxations used to solve the Steiner Tree Problem. In all cases, we choose an arbitrary terminal node $r \in R$ to serve as a root node for the Steiner tree. Such a root node creates an orientation of edges away from it. We introduce a binary variable (relaxed in the formulation) y_{ij}, which takes value of 1 if edge $\{i, j\} \in E$ is part of the solution and 0 otherwise. Similarly, we also introduce z_{ij}, a binary variable associated with an arc $(i, j) \in A$, which takes value 1 if (i, j) is part of the solution and 0 otherwise. The BCR relaxation is defined as:

$$\begin{aligned}
\min \quad & \sum_{\{i,j\} \in E} c_{ij} y_{ij} && \text{(BCR)} \\
\text{s.t.} \quad & \sum_{(i,j) \in \delta^+(V \setminus C)} z_{ij} \geq 1 && \forall C \subseteq V \setminus \{r\}, C \cap R \neq \emptyset, \\
& y_{ij} = z_{ij} + z_{ji} && \forall \{i, j\} \in E, \\
& y_{ij} \in [0, 1] && \forall \{i, j\} \in E, \\
& z_{ij} \in [0, 1] && \forall (i, j) \in A
\end{aligned}$$

It follows from the observation that in any feasible solution there must be at least one arc between two complementary sets of nodes - one containing the root node and the other containing another terminal node. The formulation is root-independent in the sense that the optimal solution value does not depend on our choice of $r \in R$ [14]. The BCR relaxation is exact (i.e., solves the IP) if $V = R$ [7], in which case STP corresponds to the minimum cost spanning tree problem.

Another commonly used relaxation is based on hypergraphs and requires a complete graph obtained by computing the metric closure of G. The first STP formulation of this type [28] was based on the observation that the problem can be formulated as that of finding a minimum-cost spanning tree of a hypergraph $(R, \mathcal{K})$ defined on the terminals. That hypergraph relies on the notion of a full-component, which is a tree with all leaves belonging to the set of terminals and all non-leaf vertices being Steiner nodes. Let $\mathcal{K}$ be the set of all full-components in our graph G, each component with associated cost C_K equal to the sum of its edge-costs. The HYP-based relaxation is defined as:

$$\min \sum_{K\in\mathcal{K}} C_K x_K \qquad \text{(HYP)} \tag{1a}$$

$$\text{s.t.} \sum_{K\in\mathcal{K}} x_K(|R\cap K\cap S|-1)^+ \le |S|-1 \quad \forall S\subseteq R, S\neq\emptyset, \tag{1b}$$

$$\sum_{K\in\mathcal{K}} x_K(|R\cap K|-1)^+ = |R|-1, \tag{1c}$$

$$x_K \ge 0 \quad \forall K\in\mathcal{K} \tag{1d}$$

Constraints (1b) are called subtour elimination constraints and enforce the absence of cycles in the solution. Constraint (1c) ensures in combination with (1b) that the solution spans all terminals.

Solving HYP to optimality is strongly NP-hard, but is solvable as an LP in polynomial time if the components are restricted to at most k terminals [28]. However, this increases the approximation ratio by factor of $1+\frac{1}{\lfloor \log_2 k\rfloor}$ [1]. Therefore, obtaining solutions with good approximation ratio requires using large k, which results in huge LPs with more than $|R|^k$ variables and constraints.

A more recent hypergraphic approach follows a chain of reasoning similar to the one used for BCR [25]. Let K^i be a directed graph obtained by directing all edges in a full-component $K\in\mathcal{K}$ towards $i\in K$. These directed hyperedges have the same cost as the undirected hyperedge K they were created from. We associate a (binary) variable x_{K^i} to each such hyperedge. Similarly to BCR, we fix a root node $r\in R$ and define $\Delta(U)=\{K^i|U\cap K\neq\emptyset,\ i\notin U\}$, a set of directed full-components leaving $U\subseteq R\setminus\{r\}$. The LP relaxation can then be written as follows:

$$\min \sum_{K\in\mathcal{K},\ i\in K} C_K x_{K^i} \qquad \text{(HYP-cut)}$$

$$\text{s.t.} \sum_{K^i\in\Delta(U)} x_{K^i} \ge 1 \quad \forall U\subseteq R\setminus\{r\},\ U\neq\emptyset, \tag{2}$$

$$x_{K^i}\ge 0 \quad \forall K^i$$

Constraints (2) say that there is at least one component crossing between the root and the other terminal nodes. This relaxation has the same optimal solution as (1a) and both are stronger than BCR [25]. Based on this relaxation, a $\ln(4)+\epsilon\approx 1.39$ randomized approximation algorithm for the Steiner tree problem was proposed by Byrka et al. [2]. Goemans et al. [15] later used it to obtain a deterministic algorithm with the same ratio proving that this is an upper bound on the integrality gap.

In Section 2, we describe how our hierarchy of extended formulations is built, starting with the commonly used multicommodity flow LP formulation for STP [29]. In Section 3, we consider several instances previously used as worst-case examples for lower bounds on integrality gap. We also elaborate on some observations related to these examples, relevant to understanding the hierarchy. Finally, we present numerical results showing that several difficult STP instances can be solved to integer optimality by using this hierarchy.

2 Stronger Path-based Extended Formulations

We introduce our extended formulation by firstly writing down the standard multicommodity flow (MCF) LP formulation for STP [29]. This formulation is equivalent to BCR [14] as it projects to the same feasible set. It models the problem in terms of multicommodity unit flows sent from the root r to every other terminal vertex $k \in R \setminus \{r\}$ (a distinct commodity k applying to every vertex k). Recall that we are given an undirected graph $G = (V, E)$, its corresponding digraph $D = (V, A)$, a cost function $c : E \to \mathbb{R}^+$ and a set of terminals $R \subseteq V$ with $r \in R$ selected as the Steiner tree root.

We introduce the following variables:

- $x_i,\ i \in V$ - models the decision to include or not a vertex i in the solution, takes value 1 if vertex i is included and 0 otherwise
- $y_{ij},\ \{i,j\} \in E$ - models the decision to include or not an edge $\{i,j\}$ in the solution, takes value 1 if the edge is included and 0 otherwise
- $z_{ij},\ (i,j) \in A$ - models the decision to include or not an arc (i,j) in the solution, takes value 1 if the arc is included and 0 otherwise
- $f^k_{ij},\ (i,j) \in A,\ k \in R \setminus \{r\}$ - equals the flow of commodity k traversing arc (i,j), takes value 1 if arc (i,j) is in the unique flow path used to deliver commodity k to its corresponding terminal and 0 otherwise

The formulation can then be written down as follows:

$$\min \sum_{\{i,j\}\in E} c_{ij} y_{ij} \qquad \text{(MCF-1)}$$

$$\text{s.t.} \quad \sum_{(j,i)\in A} f^k_{ji} = \sum_{(i,j)\in A} f^k_{ij} \qquad \forall i \in V \setminus \{k,r\}, \forall k \in R \setminus \{r\}, \tag{3a}$$

$$x_r = \sum_{(r,i)\in A} f^k_{ri} \qquad \forall k \in R \setminus \{r\}, \tag{3b}$$

$$0 \le f^k_{ij} \le z_{ij} \qquad \forall (i,j) \in A, \forall k \in R \setminus \{r\}, \tag{3c}$$

$$\sum_{(i,j)\in A} z_{ij} = x_j \qquad \forall j \in V \setminus \{r\}, \tag{3d}$$

$$y_{ij} = z_{ij} + z_{ji} \qquad \forall \{i,j\} \in E, \tag{3e}$$

$$x_i = 1 \qquad \forall i \in R, \tag{3f}$$

$$x_i, y_{ij} \in [0,1] \qquad \forall i \in V \text{ and } \{i,j\} \in E \tag{3g}$$

We call the above formulation MCF-1. The objective is a simple minimization of the total edge cost. Constraints (3a) ensure that the flow for every individual commodity is balanced at nonterminal nodes while constraints (3b) enforce the balance at the root node. As constraint (3f) ensures that variable x_r takes value 1, together with (3a) and (3b) this means that 1 unit of commodity k leaves the root and reaches its corresponding vertex $k \in R \setminus \{r\}$ (the latter condition being implied by flow balance). Inequalities (3c) tie variables z and f with z_{ij} as an upper bound on the flow f_{ij}^k of commodity $k \in R \setminus \{r\}$ traversing arc (i, j). Constraints (3d) ensure that every vertex apart from the root has precisely one incoming arc. This is not necessary for correctness of the formulation (i.e., MCF-1 is equivalent to BCR without these constraints), but serves to strengthen it. Constraint (3e) imposes an orientation for every edge in the solution and finally constraints (3g) relax the integrality constraint on x and y. Note that node variables x_i are not strictly necessary for the original multicommodity flow formulation. However, we choose to include them for the sake of constraints (3d) and possible generalization of the formulation to the Prize-Collecting Steiner Tree problem.

2.1 Extended Formulation

We now define a hierarchy of extended formulations for the Steiner tree problem. Building upon multicommodity formulation MCF-1, we keep (relaxed binary) variables x and y. The first step, similar to the work Gouveia and Telhada [16] who used this technique for the multi-weighted STP, is to simultaneously formulate the problem over every possible candidate root vertex. Therefore, variables z are redefined to include an additional root index. Similarly, we modify variables f and also rename them as w for consistency with the additional notation to be introduced later on.

- $z_{ij}^r,\ (i, j) \in A,\ r \in R$ - models the decision to include or not an arc (i, j) in the unique path used to ship commodity k from a root r to a terminal k, takes value 1 if arc is included and 0 otherwise; z can have different orientation depending on the root that is chosen
- $w_{ij}^{rk} = f_{ij}^{rk},\ (i, j) \in A,\ r \in R,\ k \in R \setminus \{r\}$ - models the flow of commodity k shipped from root r to terminal k through arc (i, j), takes value 1 if arc is on the unique commodity k path from r to k and 0 otherwise
- $u_{ij}^r,\ (i, j) \in A,\ r \in R$ - upper bounds on flow variables w_{ij}^{rk} for any commodity, takes value 1 if arc (i, j) is used to ship commodity k from root r to any terminal $k \in R \setminus \{r\}$ and 0 otherwise

We call the formulation obtained from MCF-1 by considering all possible roots MCF-1+.

Furthermore, we let $\lambda \geq 1$ denote the level of our extended formulation, which will be called MCF-λ and for $\lambda = 1$ corresponds to formulation MCF-1+. First, we fix a level $\lambda = 2$ and elaborate on the simplest improvement scenario over MCF-1. Namely, we additionally consider flows over paths of length 2 (consisting of two edges), instead of just flows over single edges. Accordingly, we define new variables w and u corresponding to flows and bounds for paths of length 2. These are added to the formulation on top of the previous variables defined for arcs. With every consecutive level of the hierarchy, it is necessary to define new path variables w and u with increasing number of indices. While this unorthodox practice might cause some confusion, we use it to simplify the notation of the general MCF-λ formulation. Variables for MCF-2 are defined as follows:

- $w^{rk}_{lij},\ (l,i),(i,j) \in A,\ r \in R,\ k \in R \setminus \{r\}$ - models the flow of commodity k shipped from root r to terminal k through a directed path (l,i,j) of length 2, takes value 1 if path on the unique commodity k path from r to k and 0 otherwise
- $u^{r}_{lij},\ (l,i),(i,j) \in A,\ r \in R$ - upper bound on flow variables w^{rk}_{lij} for any commodity, takes value 1 if path (l,i,j) is used to ship commodity k from root r to any terminal $k \in R \setminus \{r\}$ and 0 otherwise

These variables are illustrated in Figure 1.

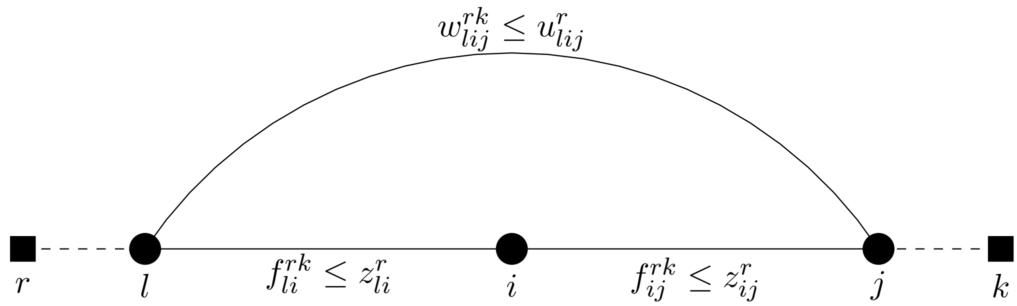


Figure 1: Graphical representation of new variables for a case of length two paths.

Furthermore, for notation purposes, we introduce two sets:

- $\mathcal{A}^{rk}_2 = \{(l,i,j)|(l,i),(i,j) \in A, l \neq k, i \neq r, i \neq k, j \neq r, j \neq l\},\ r \in R,\ k \in R \setminus \{r\}$, set of length two paths in graph G, that might be used to ship commodity k from root r to terminal k, meaning that r can only be the first node and k can only be the final node (though they are not necessarily included in the length two path)
- $\mathcal{A}^{r}_2 = \{(l,i,j)|(l,i),(i,j) \in A, i \neq r, j \neq r, j \neq l\},\ r \in R$ - set of length two paths in graph G, that might be used to ship commodities from root r to every other terminal vertex (though r is not necessarily included in the length two path)

We now write down the extended formulation for this case:

$$\min \sum_{\{i,j\}\in E} c_{ij} y_{ij} \qquad \text{(MCF-2)}$$

$$\text{s.t.} \sum_{(l,i,j)\in\mathcal{A}_2^{rk}} w_{lij}^{rk} = \sum_{(i,j,l)\in\mathcal{A}_2^{rk}} w_{ijl}^{rk} \quad \forall (i,j)\in\{A|i\neq r, j\neq k\}, \forall (r,k)\in\mathcal{RK}, \tag{4a}$$

$$x_r = \sum_{(r,i,j)\in\mathcal{A}_2^{rk}} w_{rij}^{rk} \qquad \forall (r,k)\in\{\mathcal{RK}|(r,k)\notin A\}, \tag{4b}$$

$$x_r = \sum_{(r,i,j)\in\mathcal{A}_2^{rk}} w_{rij}^{rk} + w_{rk}^{rk} \qquad \forall (r,k)\in\{\mathcal{RK}|(r,k)\in A\}, \tag{4c}$$

$$w_{ri}^{rk} = \sum_{(r,i,j)\in\mathcal{A}_2^{rk}} w_{rij}^{rk} \qquad \forall (r,i)\in\{A|i\neq k\}\forall (r,k)\in\mathcal{RK}, \tag{4d}$$

$$w_{ri}^{rk} \leq u_{ri}^{r} \qquad \forall (r,i)\in A, \forall (r,k)\in\mathcal{RK}, \tag{4e}$$

$$w_{lij}^{rk} \leq u_{lij}^{r} \qquad \forall (l,i,j)\in\mathcal{A}_2^{rk}, \forall (r,k)\in\mathcal{RK}, \tag{4f}$$

$$z_{ij}^{r} = \sum_{l:(l,i,j)\in\mathcal{A}_2^{r}} u_{lij}^{r} \qquad \forall (i,j)\in\{A|i\neq r\}, \forall r\in R, \tag{4g}$$

$$z_{ij}^{r} = \sum_{l:(l,i,j)\in\mathcal{A}_2^{r}} u_{lij}^{r} + u_{ij}^{r} \qquad \forall (i,j)\in\{A|i= r\}, \forall r\in R, \tag{4h}$$

$$\sum_{(i,j)\in A} z_{ij}^{r} = x_j \qquad \forall j\in V\setminus\{r\},\ \forall r\in R, \tag{4i}$$

$$y_{ij} = z_{ij}^{r} + z_{ji}^{r} \qquad \forall \{i,j\}\in E,\ \forall r\in R, \tag{4j}$$

$$x_i = 1 \qquad \forall i\in R, \tag{4k}$$

$$x_i, y_{ij}\in[0,1] \qquad \forall i\in V \text{ and } \{i,j\}\in E \tag{4l}$$

Note that the constraints (4i)-(4l) have the same role in formulation MCF-2 as constraints (3d)-(3g) do in MCF-1. The flow balance constraints on nodes (3a) are now replaced by flow balance constraints on arcs (4a), meaning that one now balances flows through individual arcs. However, this generalization only applies to arcs which are not adjacent to the root or their accompanying terminals. Constraints (4b) and (4c) correspond to (3b) as they balance the flow over all arcs pointing out of the root. This can include paths of length 2 or single arcs if there is an edge directly linking the root to an accompanying terminal. A tightening occurs at constraints (4d)-(4h), which correspond to constraints (3c) in MCF-1. The first of these constraints makes sure that length two paths with roots as their initial vertices are properly defined. The second and third constraints impose bounds on flows over length-1 and length-2 paths respectively, as illustrated in Figure 1. Finally, the last two constraints relate the arc capacity to path capacities.

We now generalize this approach by allowing paths of length up to $\lambda \geq 2$. We define new variables and sets in addition to the ones introduced before the MCF-2 formulation:

- w_p^{rk} - models the flow of commodity k shipped from a root r to a terminal $k \in R\backslash\{r\}$

through a path p of length at most λ, takes value 1 if the path is part of the integer solution and 0 otherwise

- u_p^r - can be interpreted as a switch on the use or not of path p from root r, takes value 1 if the path is included in the final solution and 0 otherwise. Another interpretation is that it takes value 1 if path p is part of the arborescence rooted at r and 0 otherwise

- $\mathcal{A}_l^{rk} = \{(j_1, j_2, \ldots, j_{l+1}) | (j_i, j_{i+1}) \in A \ \forall i = \{1, 2, \ldots, l\}, j_i \neq r \ \forall i = \{2, 3, \ldots, l+1\}, j_i \neq k \ \forall i = \{1, 2, \ldots, l\}, i \neq l \iff j_i \neq j_l \ \forall i, j = \{1, 2, \ldots, l+1\}\}$ - set of paths of length l in graph G, that may be used to ship commodity k from root $r \in R$ to terminal $k \in R \setminus \{r\}$, meaning that r can only be the first node and k can only be the final node (though they are not necessarily included in the path)

- $\mathcal{A}_l^{r} = \{(j_1, j_2, \ldots, j_{l+1}) | (j_i, j_{i+1}) \in A \ \forall i = \{1, 2, \ldots, l\}, j_i \neq r \ \forall i = \{2, 3, \ldots, l+1\}, i \neq l \iff j_i \neq j_l \ \forall i, j = \{1, 2, \ldots, l+1\}\}$ - set of paths of length l in graph G, that may be used to ship commodities from a root $r \in R$ to every other terminal vertex, meaning that r can only be the first node (though it is not necessarily included in the path)

- $P_p^{rk} = \{n \in \mathcal{A}_\lambda^{rk} | n = (\ldots, p)\} \cup \{\cup_{l=1}^{\lambda-1} \{n \in \mathcal{A}_l^{rk} | n = (r, \ldots, p)\}\}$ - set of paths directed from a root $r \in R$ to a terminal $k \in R \setminus \{r\}$, ending in a path p or a vertex p and of length λ or shorter if beginning at r

- $P_p^{r} = \{n \in \mathcal{A}_\lambda^{r} | n = (\ldots, p)\} \cup \{\cup_{l=1}^{\lambda-1} \{n \in \mathcal{A}_l^{r} | n = (r, \ldots, p)\}\}$ - set of paths directed from root $r \in R$, ending in a path p or a vertex p and of length λ or shorter if beginning at r

- $Q_p^{rk} = \{n \in \mathcal{A}_\lambda^{rk} | n = (p, \ldots)\} \cup \{\cup_{l=1}^{\lambda-1} \{n \in \mathcal{A}_l^{rk} | n = (p, \ldots, k)\}\}$ - set of paths directed from root $r \in R$ to a terminal $k \in R \setminus \{r\}$, beginning with a path p or a vertex p and of length λ or shorter if ending in k

Variables x, y and z remain as previously defined. The general MCF-λ extended formulation can be written as:

$$\min \sum_{\{i,j\}\in E} c_{ij} y_{ij} \qquad \text{(MCF-}\lambda\text{)}$$

$$\text{s.t.} \quad \sum_{q\in P_p^{rk}} w_q^{rk} = \sum_{q\in Q_p^{rk}} w_q^{rk} \quad \forall p \in \{\mathcal{A}_{\lambda-1}^{rk} | r \notin p, k \notin p\}, \forall (r,k) \in \mathcal{RK}, \tag{5a}$$

$$x_r = \sum_{p\in Q_r^{rk}} w_p^{rk} \quad \forall (r,k) \in \mathcal{RK}, \tag{5b}$$

$$w_p^{rk} = \sum_{q\in Q_p^{rk}\cap \mathcal{A}_{l+1}^{rk}} w_q^{rk} \quad \forall p \in \{\mathcal{A}_l^{rk} | r \in p, k \notin p\}, \forall (r,k) \in \mathcal{RK}, \quad \forall l = \{1,2,\ldots,\lambda-1\}, \tag{5c}$$

$$w_p^{rk} \le u_p^r \quad \forall p \in \mathcal{A}_\lambda^{rk}, \forall (r,k) \in \mathcal{RK}, \tag{5d}$$

$$w_p^{rk} \le u_p^r \quad \forall p \in \{\mathcal{A}_l^{rk} | r \in p\}, \forall (r,k) \in \mathcal{RK}, \quad \forall l = \{1,2,\ldots,\lambda-1\}, \tag{5e}$$

$$z_{ij}^r = \sum_{p\in P_{ij}^r} u_p^r \quad \forall (i,j) \in A, \forall r \in R, \tag{5f}$$

$$(4i) - (4l)$$

We keep the last four constraints of MCF-2. Constraints (5a) enforce flow balance for paths of length $\lambda - 1$. They play the same role for paths of length $\lambda - 1$ in MCF-λ as the constraints (4a) in MCF-2. In simple terms they enforce that, given a path, the sum of flows into it is equal to sum of flows out of it. Root node flow balance is now enforced in (5b) as a sum of flows along paths that may eventually be longer that those in (4b) and (4c) and that ship the required one unit of commodity k from a root r to a terminal $k \in R \setminus \{r\}$. Constraints (5d) and (5e) represent generalized constraints (4e) and (4f) and impose flow bounds on all paths up to length λ. In a similar fashion to constraint (5b), constraint (5f) relates the capacity of an arc (i,j) to a sum of capacities of paths of length at most λ. Constraints (5c) now provide proper definition of flows on short paths of lengths 1 through $\lambda - 1$, which are adjacent to the root node. This is important in order to allow a proper calculation of capacity z_{ij}^r near root r through constraints (5d), (5e) and (5f). Note that only the following flow path variables (and corresponding path capacities u) have to be defined:

- variables representing flow on paths of length λ,
- variables representing flow on shorter paths that begin at root r.

It should be noted that for a fixed level λ, the problem description contains a polynomial number of variables and constraints. The highest order of magnitude for them is determined by the number of paths of length λ that may eventually exist in the graph. If we consider a complete graph, then there are $\mathcal{O}(|V|^{\lambda+1})$ such paths. Given that we consider directed paths for all pairs of terminals, one would thus have $\mathcal{O}(|V|^{\lambda+1}|T|^2)$ variables and constraints.

For some graph instances, especially those containing longer paths, MCF-λ with small parameter λ can fail to match the results obtained by HYP or MCF-n. We discuss this situation in more detail in the next section in a paragraph associated with Figure 5, which presents an instance of a graph that causes such problems for MCF-2. It can be addressed by increasing λ, which can significantly increase the size of the formulation, but the result can be made arbitrarily bad by subdividing the path with more vertices. Alternatively, we can replace graph G by its metric closure and introduce an additional constraint for each Steiner (non-terminal) node:

$$\sum_{j:(i,j)\in A} z^r_{ij} \geq 2x_i \qquad \forall i \in V \setminus R \quad \forall r \in R. \tag{6}$$

This means that each Steiner node, if included in the final solution, has at least 2 outgoing edges in addition to the constraint saying that it has at most one incoming edge. Equivalently, we could write down a constraint saying that each Steiner node is either of degree 0 or at least 3. This approach also increases the size of the problem, but the number of variables and constraints remains of the same order of magnitude. Metric closure is also required for hypergraphic formulations [23], so MCF-λ does not suffer a disadvantage in comparison.

Finally, we present an extended formulation of maximum level (we call it MCF-n), i.e., a formulation that involves the longest paths possible in the graph.

$$\min \sum_{\{i,j\}\in E} c_{ij}y_{ij} \qquad \text{(MCF-n)}$$

$$\text{s.t.} \quad x_r = \sum_{p\in Q^{rk}_r \cap P^{rk}_k} w^{rk}_p \quad \forall (r,k) \in \mathcal{RK}, \tag{7a}$$

$$w^{rk}_p = \sum_{q\in Q^{rk}_p \cap \mathcal{A}^{rk}_{l+1}} w^{rk}_q \;\; \forall p \in \{\mathcal{A}^{rk}_l | r \in p, k \notin p\}, \forall (r,k) \in \mathcal{RK}, \quad \forall l = \{1,2,\ldots,n-1\}, \tag{7b}$$

$$w^{rk}_p \leq u^r_p \qquad \forall p \in \mathcal{A}^{rk}_l, \forall (r,k) \in \mathcal{RK}, \forall l = \{1,2,\ldots,n\}, \tag{7c}$$

$$z^r_{ij} = \sum_{p\in P^r_{ij}} u^r_p \qquad \forall (i,j) \in A, \forall r \in R, \tag{7d}$$

$$(4i) - (4l)$$

This differs from MCF-λ only in one aspect, namely constraints (5a) and (5b) are replaced by (7a), that only involve balance constraints for every individual root vertex. Namely, we sum all the flows in k-commodity paths of any length that begin at a root $r \in R$ and end at a terminal $k \in R \setminus \{r\}$. Similarly to MCF-λ, flow balance at commodity terminals is preserved automatically due to constraints (5a) and (5b). In addition to all flows along root-to-terminal paths, also shorter paths beginning at a root are included to facilitate strengthening through bounds u in constraints (7b),(7c) and (7d). In practice, even a single flow balance constraint (7a) that splits the flow among all possible paths from root to commodity makes the problem prohibitively expensive to solve.

We now present two theoretical results related to our hierarchy of extended formulations.

Proposition 1. *MCF-$(\lambda+1)$ is at least as strong as MCF-λ for $\lambda \geq 1$.*

Proof. To prove this it suffices to show that any MCF-$(\lambda+1)$ solution can be converted to a MCF-λ solution with the same cost. Note that constraints (4i)-(4l) remain the same for all levels of $\lambda \geq 1$. For the remaining constraints, it is necessary to sum the flows and bounds of paths of length $\lambda+1$ to obtain the flows and bounds on paths of length λ. This is carried out in the style of constraints (5c) as follows:

$$w_p^{rk} = \sum_{q \in Q_p^{rk} \cap \mathcal{A}_{\lambda+1}^{rk}} w_q^{rk} \quad \forall p \in \{\mathcal{A}_\lambda^{rk} | k \notin p\}, \forall (r,k) \in \mathcal{RK}$$

$$w_p^{rk} = \sum_{q \in P_p^{rk} \cap \mathcal{A}_{\lambda+1}^{rk}} w_q^{rk} \quad \forall p \in \{\mathcal{A}_\lambda^{rk} | r \notin p\}, \forall (r,k) \in \mathcal{RK}$$

$$u_p^{r} = \sum_{q \in Q_p^{r} \cap \mathcal{A}_{\lambda+1}^{r}} u_q^{r} \quad \forall p \in \{\mathcal{A}_\lambda^{r} | k \notin p\}, \forall r \in R$$

$$u_p^{r} = \sum_{q \in P_p^{r} \cap \mathcal{A}_{\lambda+1}^{r}} u_q^{r} \quad \forall p \in \{\mathcal{A}_\lambda^{r} | r \notin p\}, \forall r \in R$$

This reasoning can be applied in a straightforward way to constraints (5a), (5b) and (5f). For (5c) and (5d) and (5e), all the necessary constraints for the λ-level formulation are already included and we remove the ones related to paths of length $\lambda+1$. □

Proposition 2. *There exist instances of graphs, for which MCF-$(\lambda+1)$ is stronger than MCF-λ.*

Proof. We refer to the next section for examples of graph instances where there is an improvement (e.g., graph in Figure 9). □

3 Analysis of bounds for worst-case instances of HYP and BCR

In this section, we present empirical results for our hierarchy of extended formulations and report on computational experiments carried out with use of the ZIMPL mathematical modeling language [21] and Gurobi LP solver [17]. In all the following figures illustrating various graph instances, squares and circles are used to represent terminal nodes and Steiner vertices respectively. Unless shown otherwise, the edge weights are 1.

First, consider the Goemans [13] family of graphs, which can be defined as follows. Define a single 'origin' node n_0 and N additional nodes $n_1, \ldots, n_N$, all of which are terminals. We then add Steiner nodes m_i with incident edges $\{n_0, m_i\}$ and $\{n_i, m_i\}$ for every $i = 1, 2, \ldots, N$. Each of these edges has cost 2. Additional Steiner vertices p_{ij} are also introduced with cost 2 edges $\{p_{ij}, m_i\}$ and $\{p_{ij}, m_j\}$ incident to them. At last, vertices q_{ij} are also introduced with incident edges $\{q_{ij}, m_i\}$, $\{q_{ij}, m_j\}$ and $\{q_{ij}, p_{ij}\}$ of cost 1 for every applicable pair i, j with $i, j = 1, 2, \ldots, N$, $i \neq j$. This is illustrated in Figure 2a. In our tests we use the simplest graph of this family, defined for $N = 2$ (Figure 2b), as well as two modified graphs with additional terminals as illustrated in Figures 2c and 2d.

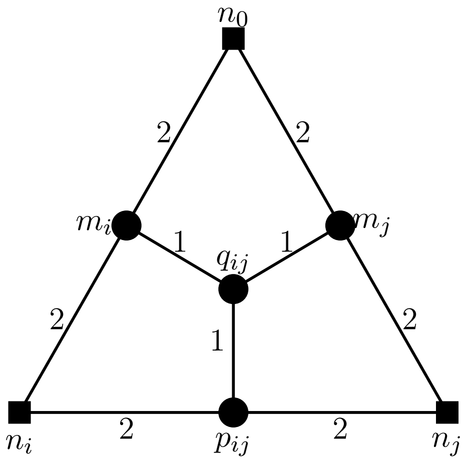
BCR obtains fractional solutions for the graphs displayed in Figures 2b and 2c with a solution value of 7.5 in both cases. These fractional solutions assign flows of 0.5 at every edge of the graph and were used to establish a $16/15 \approx 1.07$ integrality gap for BCR. In case of the hypergraphic formulation, this gap is closed for 2b, but not for 2c, where a 16/15 integrality gap is still present. Considering our extended formulation, the fractional solutions for these graphs are eliminated by MCF-2, which reaches the integer optimal value of 8. A similar result is obtained for the instance (2d), where BCR attains a fractional solution value of 5.5 with a lower bound on integrality gap of 1.09. HYP closes this gap by obtaining the integer solution 6. This result is also attained by MCF-2. However, this is also an example showing that, contrary to the case of BCR, the choice of the root cannot be arbitrary. If only a single root is considered, then the optimal integer solution is only obtained by MCF-1 in case of one of the degree 2 terminals is taken as the root node. For the remaining terminal, only the fractional solution of BCR will be obtained. This justifies our choice of including all possible roots in the formulation. The solutions are summarized in Table 1, together with three graph instances introduced later on in this section. The results for BCR are matched by MCF-1 and MCF-1+.

Instance	Nodes	Terminals	Edges	BCR	HYP	MCF2	MCF3	MCF4	OPT
2b	7	3	9	7.5	8	8	8	8	8
2c	7	4	9	7.5	7.5	8	8	8	8
2d	8	4	10	5.5	6	6	6	6	6
5	16	3	18	30	32	31.5	32	32	32
7	15	8	35	8.75	8.75	9.40625	9.40625	9.40625	10
9	22	10	39	15	15.75	15.75	16.5	16.5	17

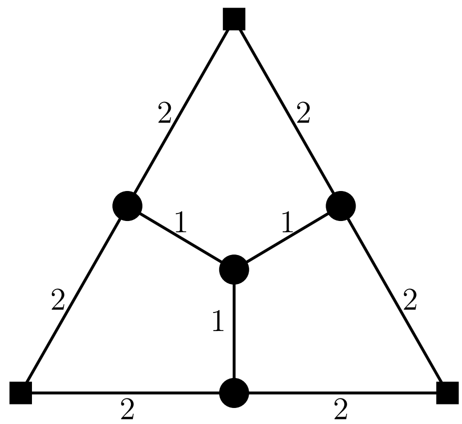
Table 1: Comparison of optimal objective function values for standard and extended formulations on selected graph instances.

We now present more detailed information on the solutions obtained by the different formulations on instances 2b, 2c and 2d. Firstly, in Figure 3, solutions are displayed for instance 2b. The BCR solution allows for the flow to be sent through the central node, which is not allowed by HYP or MCF-2. To be precise, BCR assigns 0.5 units to each edge in the graph, while MCF-2 and HYP lead to an integer solution, which follows the outer edges.

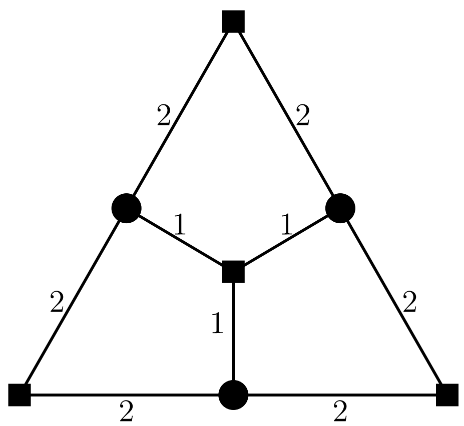
In Figure 4, the solutions to instance 2c present the case in which HYP matches the BCR solution and is surpassed by MCF-2. The BCR solution is similar to the



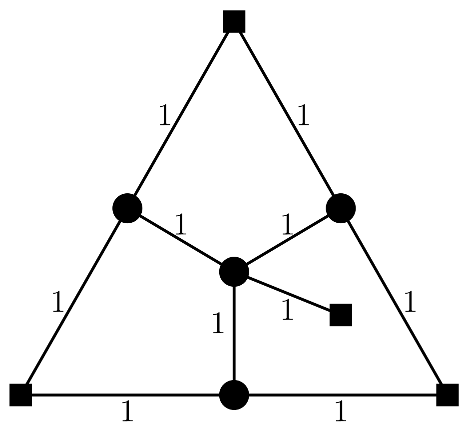
(a) Segment of a graph belonging to the family of graphs proposed by Goemans.



(b) An instance for $N = 2$ with 3 terminals.



(c) A modified instance with an additional 'central' terminal node.

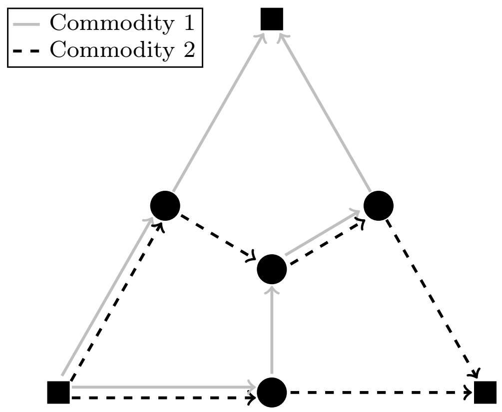


(d) A modified instance with an additional terminal node attached to the central vertex [9].

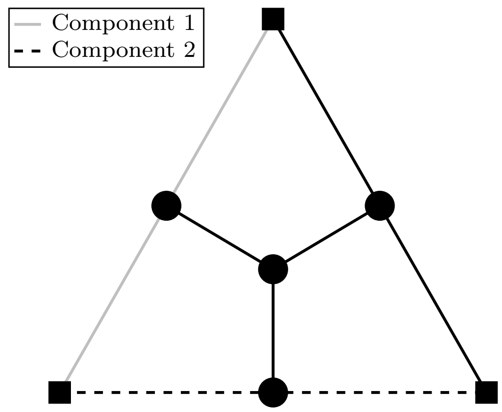
Figure 2: Instances of graphs based on the family of graphs proposed by Goemans [13].

previous case and assigns 0.5 capacity to each line. Using HYP leads to a choice of three components, each with 0.5 weight. On the other hand, using MCF-2 leads to an integer solution.

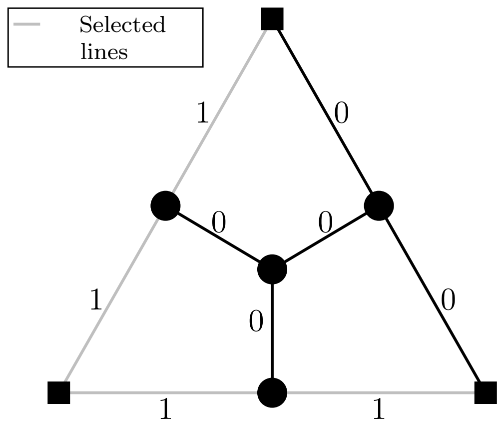
In Figure 5, we present an instance of a graph that serves as an example justifying the use of metric closure and constraint (6). It is obtained by taking graph 2b and replacing the inner edges with paths consisting of 4 edges. Solving the graph instance



(a) Solution using BCR: For each commodity there are 2 paths of flow, one along the outer edges and the other through the center. Each path carries 0.5 units of flow.



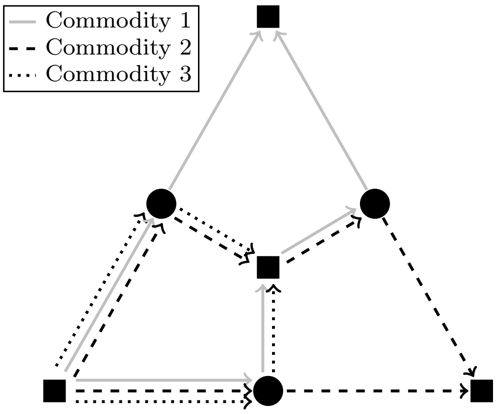
(b) Solution using HYP: Two components joining the terminals along outer edges are chosen, one for each commodity.



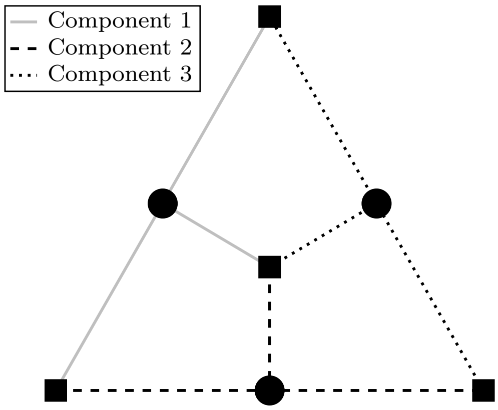
(c) Solution using MCF-2 in terms of edge variables y. Only four outer edges are picked, such that they provide a solution similar to HYP.

Figure 3: Sample solutions of instance 2b with BCR, HYP and MCF-2 formulations.

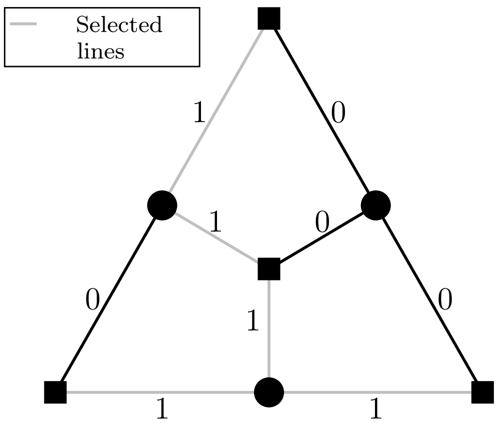
with BCR and HYP yields objective function values of 30 and 32 respectively (the last of which matches the integer optimal solution). MCF-2 attains a value of 31.5, which is better than that for BCR, but does not match the one attained by HYP. This can be



(a) Solution using BCR: For each commodity there are two paths of flow. Each path carries 0.5 units of flow.



(b) Solution using HYP makes use of three components, each with weight 0.5.



(c) Solution using MCF-2 in terms of edge variables y. Edges are chosen such that the flow along one outer edge is replaced by flow through the central terminal.

Figure 4: Sample solutions of instance 2c with BCR, HYP and MCF-2 formulations.

remedied either by using MCF-3, which matches the result provided by HYP, or by using the metric closure and constraint 6 as described in the previous section. Figure 6 shows MCF-2 and MCF-3 solutions for this instance and what fractional flow is eliminated by

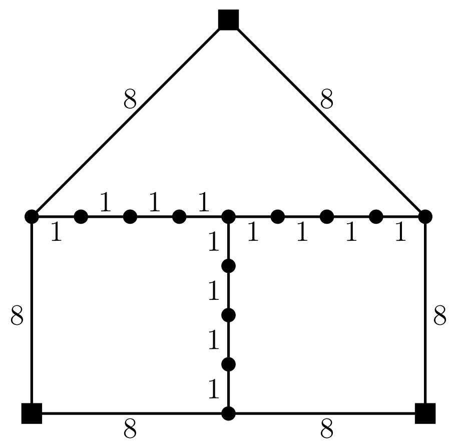


Figure 5: Instance of a modified Goemans graph (2a) showing issues in MCF-2 performance.

the stronger formulation.

Skutella's graph shown in Figure 7 is a basis for the current lower bound of $8/7 \approx 1.14$ on the integrality gap for the HYP formulation [2]. It consists of 15 vertices divided into three layers - the first one including a terminal, which is originally intended as a root node, then a layer of seven Steiner nodes and finally a layer of 7 terminals. The initial terminal is connected to every Steiner node. The edges between second and third layers follow the rule that each node has four adjacent edges and each pair of nodes from one layer is connected to exactly six different nodes in the other layer. The optimal fractional solution given by both BCR and HYP sends 0.25 units of flow along every edge, giving an objective function value of 8.75 with the integer optimum being 10. The integer optimal solution uses 3 Steiner nodes from the second layer to connect to the lowest layer.

Skutella's graph can be extended to an arbitrary number of layers. Given an original graph, we can add a new layer to it as follows: take each terminal node at the bottom layer, turn it into a Steiner node and duplicate it 6 times (for a total of 7 identical nodes) together with edges to the nodes in the layer above. Now, for each set of 7 newly obtained nodes, we create 7 new terminals one layer lower and join then to the above with the same pattern of edges from original Skutella's graph. We can repeat this process as many times as necessary to obtain more and more complicated graphs with increasing lower bound values for BCR. For a single additional layer, this approach is pictured in Figure 8. It was proven by Byrka et al. [2] that the upper bound on the integrality gap for BCR approaches $36/31 \approx 1.161$, when the number of layers for this graph instance goes to infinity. The original (3-level) Skutella graph serves as an example of the fact that strict improvement is not guaranteed in our hierarchy. MCF-2 attains a fractional solution of $9.40625 = 301/32$ for it, resulting in an integrality gap of $320/301 \approx 1.06$. This solution assigns capacities of $5/32$ and $19/64$ to the upper set of edges and the lower set of edges

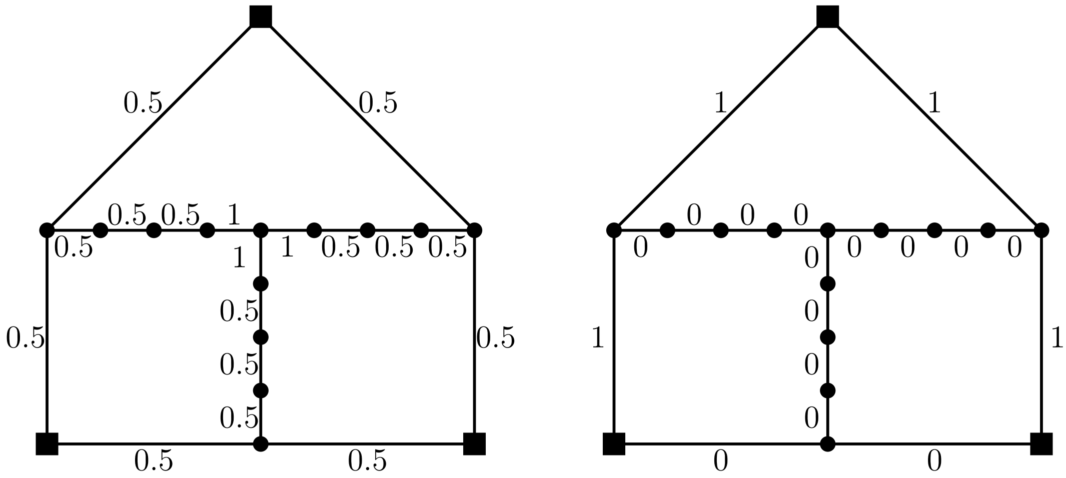


Figure 6: The MCF-2 (left) and MCF-3 (right) solutions in terms of y variables to the instance presented in Figure 5. The higher level formulation eliminates flow joining and splitting at the center node.

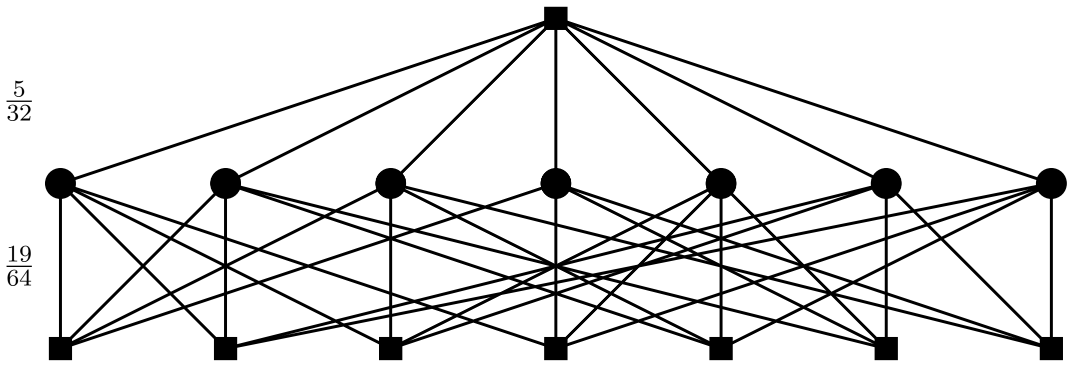


Figure 7: Skutella's graph, an example of 8/7 integrality gap for HYP formulation. All edges have weight 1. Numbers on the left side show the flow limits provided by our hierarchy of extended formulations.

respectively. Moreover, this graph instance is also an example that choosing the largest possible λ in a graph does not produce the convex hull of the integer problem.

Proposition 3. *The MCF-n formulation, where n is the number of nodes in G, does not characterize the convex hull of feasible solutions to the Steiner tree problem.*

Proof. Let ξ_{14} be value of the optimal solution to MCF-14, which is an extended formulation of maximum level for Skutella's graph. If we restrict the problem such that flows on

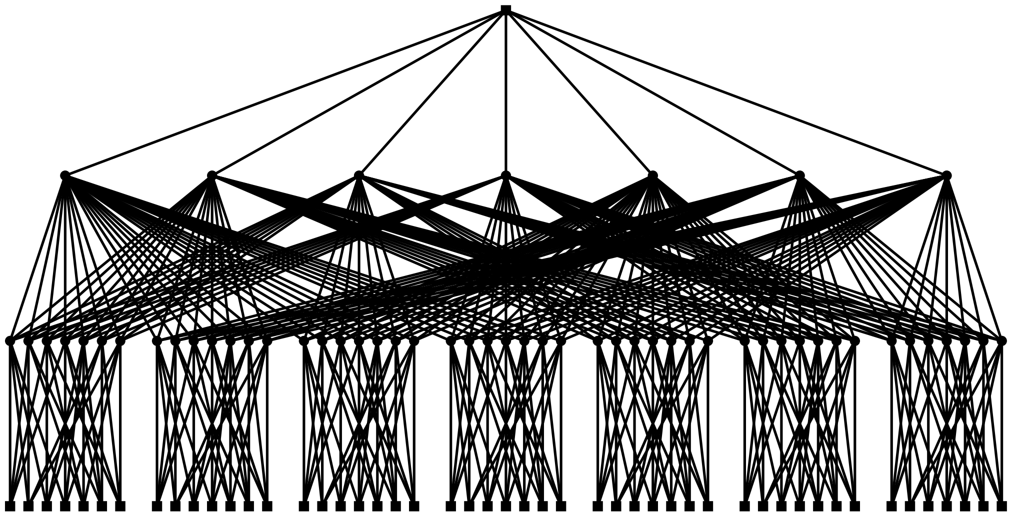


Figure 8: Elevated Skutella's graph. All edges have weight 1.

paths of length larger than 4 are set to 0, we can obtain the value of the optimal solution to the restricted problem, which we call ξ^*_{14}. Since our problem is a minimization one, we have $\xi_{14} \leq \xi^*_{14}$. Note that, considering the constraint sets for MCF-14 and MCF-4, $\xi^*_{14} = \xi^*_4$, where ξ^*_4 is value of the optimal solution to MCF-4 with flow on paths that do not start at r set to 0. Finally, observe that there exists an optimal solution to MCF-4 that meets this criterion, so $\xi^*_4 = \xi_4$, where ξ_4 is value of the optimal solution to MCF-4. Hence, for Skutella's graph instance we have

$$\xi_{14} \leq \xi^*_{14} = \xi^*_4 = \xi_4.$$

From Proposition 1 we know that $\xi_{14} \geq \xi_4$ and therefore $\xi_{14} = \xi_4$. This proves for that graph instance that optimal solution values are the same for MCF-λ with $4 \leq \lambda \leq 14$ and are equal to 9.40625, which is not the optimal integer solution value. Hence, the extended formulation for the maximum path length possible does not characterize the convex hull of feasible solutions to STP. □

We can also apply our approach to the graph presented by Byrka et al. [2]. For that graph, all edge weights are also set to 1. There are now 4 layers of vertices, one terminal at the top layer, which is connected to three Steiner nodes in the second layer. The third and fourth layers contain 9 nodes each, Steiner and terminal nodes respectively. The remaining connections are shown in Figure 9. The optimal integer solution to this instance is 17, while BCR and HYP provide fractional solutions of 15 and 15.75 respectively. The latter one is matched by MCF-2 and improved upon by MCF-3, which attains 16.5. The weakest fractional solution, given by BCR, has bound of 0.25 for flows in the middle of the graph (joining second and third layers) and 0.5 for all other edges. HYP on the other hand assigns flows of 1/12 over all 27 symmetric components [2]. Our MCF-2

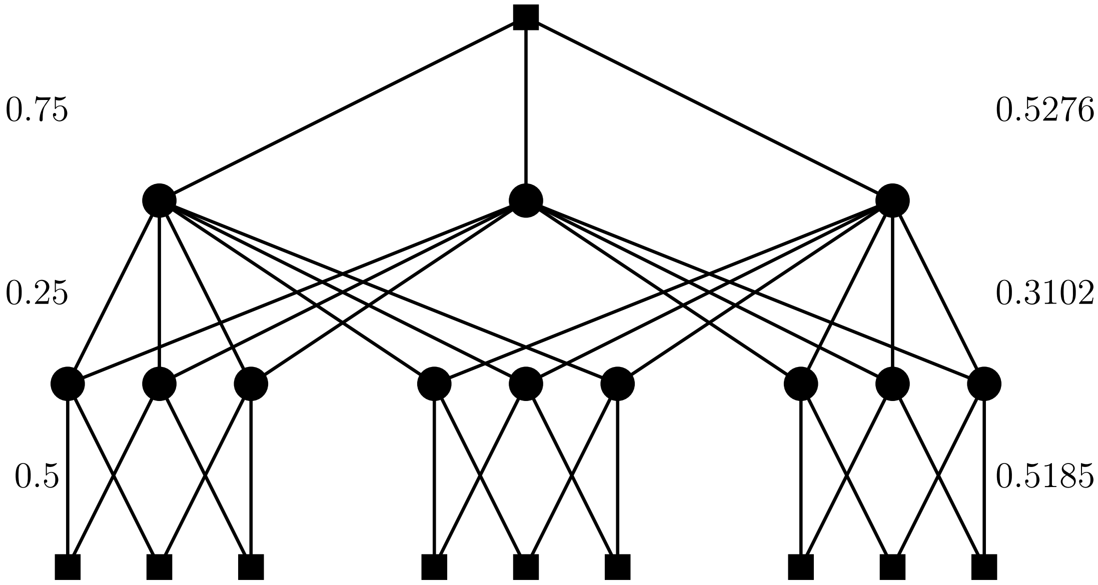


Figure 9: A different instance of a graph showing improvement between consecutive extended formulation levels. Numbers on the left side show the flow limits provided by MCF-2 and numbers on the right side show flow limits provided by MCF-3.

formulation assigns capacity of 0.75 for the top edges, 0.25 for the middle set and 0.5 for the bottom set of the edges. For the case of MCF-3, these capacities are approximately 0.5276, 0.3102 and 0.5185 respectively.

For the sake of completeness, we present an overview of lower and upper bounds in Figure 10.

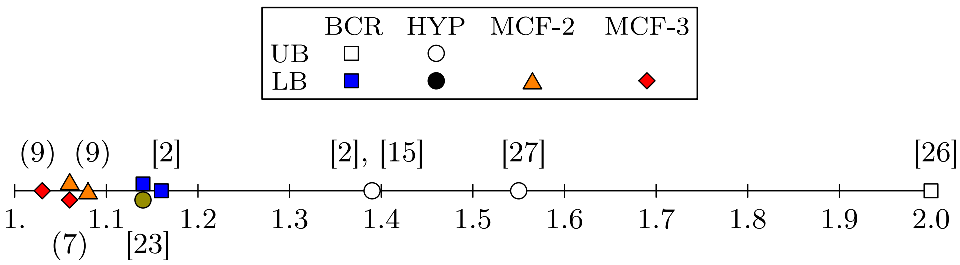


Figure 10: Comparison of integrality gap bounds for different formulations of STP.

4 Numerical Results

In this section, we present numerical results obtained by applying our extended formulation to some difficult test instances. We used ZIMPL [21] as the modeling language and Gurobi 7.5 [17] as the linear programming solver with all presolve features turned off. Due to increasing power of current solvers, test instances often need to be specially designed to be difficult. For example, our initial tests on randomly generated instances for graphs up to 40 nodes were directly solved to integer optimality by MCF-1. To provide meaningful results, we applied our extended formulation to several difficult instances from the I080 test set in the SteinLib library [22]. Tests were performed on a computer with a 2.4 GHz CPU and 8 GB of RAM. A time limit of 2 CPU hours was imposed on every run of the algorithm. Table 2 shows, from left to right, the name of each instance, its size (number of edges and terminals), the optimal integer solution value, the objective function values attained by MCF-1 and MCF-2 and corresponding CPU times.

Instance	|E|	|R|	Opt	MCF-1		MCF-2		Closed
				Value	Time (s)	Value	Time (s)	gap
i080-044	632	6	1366	1329	3	1366	887	100.00%
i080-111	350	8	2051	2025.44	25	2049.99	596	96.05%
i080-143	632	8	1767	1755.29	44	1767	2395	100.00%
i080-212	350	16	3677	3676.79	2224	3677	6030	100.00%
i080-213	350	16	3678	3650.33	1056	3677.84	7200	99.42%
i080-214	350	16	3734	3696.07	1250	3724.89	7200	75.98%
i080-215	350	16	3681	3672.91	1692	3681	5735	100.00%
i080-235	160	16	4487	4480.5	115	4487	452	100.00%
i080-305	120	20	5932	5888.5	82	5894.5	288	13.79%
i080-331	160	20	5226	5206.5	218	5213	589	33.33%
i080-332	160	20	5362	5339.5	194	5362	683	100.00%

Table 2: Results of computational experiments on I080 instances with 80 nodes each.

While the computation times cannot compare to current state-of-the-art algorithms for STP, they give us an indication of the computational effort necessary to solve higher-level extended formulations from our hierarchy. In general, a significant increase in computation time is expected due to the larger size of the problem. On the other hand, we observe that in all cases a substantial fraction of the remaining gap was closed by MCF-2 and in most cases that fraction was very high. In particular, 6 out 11 instances were solved to optimality by MCF-2. Moreover, applying the MCF-3 formulation to two of the worst performing instances (in terms of the percentage of the integrality gap closed), i080-305 and i080-331, yields optimal solutions values of 5929.6 (94.48% integrality gap reduction) and 5226 (100% integrality gap reduction) respectively. On a practical side however, the demand for memory and computational power highly increased with size of the problem and 2 instances timed out, even when solving their MCF-2 formulation.

5 Conclusion

In this paper we proposed a hierarchy of extended formulations of increasing strength. We have shown that our extended formulations improve the solutions currently provided by hypergraphic and bidirected cut formulations to all graph instances we have tested, such as Skutella's graph and the Goemans family of graphs. Since these are used to show the worst-case lower bound for the integrality gap, it suggests that MCF-λ might lead to better approximation algorithms, which can be more efficient than the computationally intensive hypergraphic formulations. On the other hand, the computational effort required to solve MCF-λ to optimality increases with each consecutive level of our extended formulation and hence only smaller λ are practical, especially as increasing this parameter is not guaranteed to improve the bound.

Our framework is consistent with problems related to STP, such as the prize-collecting Steiner tree problem or the Steiner forest problem, and hence can be adapted to them, among others. Still, further research is necessary to consider possible improvement over the trivial upper bound of 2, detailed comparison of MCF-2 and HYP, behavior on some special graphs instances (e.g., bipartite and quasi-bipartite graphs), development of tractable MCF-λ formulations and overall improvement in deciding which level λ is the most suitable. Higher levels of this hierarchy can also be used to develop valid inequalities for lower levels, such as MCF-1, corresponding to BCR.

One other possible approach is to consider the path-based hypergraphic formulation. By defining new variables for 'paths of components', this might lead to further quality improvements due to HYP being stronger than BCR. However, the path-based approach also increases computational complexity, which in turn means that the problems will become intractable faster than the standard HYP approach.

References

[1] A. Borchers and D.Z. Du, *The k-Steiner ratio in graphs*, SIAM J. Comput. **26** (1997), 857–869.

[2] J. Byrka, F. Grandoni, T. Rothvoss, and L. Sanità, *Steiner tree approximation via iterative randomized rounding*, J. ACM **60** (2013), 6:1–6:33.

[3] D. Chakrabarty, J. Könemann, and D. Pritchard, *Hypergraphic LP relaxations for Steiner trees*, Integer Programming and Combinatorial Optimization, Springer Berlin Heidelberg, 2010, pp. 383–396.

[4] M. Chlebík and J. Chlebíková, *Approximation hardness of the Steiner tree problem on graphs*, Algorithm Theory — SWAT 2002, Springer Berlin Heidelberg, 2002, pp. 170–179.

[5] S. Chopra and C.Y. Tsai, "*Polyhedral approaches for the Steiner tree problem on graphs*," *Steiner Trees in Industry*, X.Z. Cheng and D.Z. Du (eds.), Springer US, Boston, MA, 2001, pp. 175–201.

[6] S.V. Daneshmand, Algorithmic approaches to the Steiner problem in networks, Ph.D. thesis, Universität Mannheim, 2004.

[7] J. Edmonds, *Optimum branchings*, J. Res. National Bureau Standards **71B** (1967), 233–240.

[8] J. Edmonds, *Matroids and the greedy algorithm*, Math. Program. **1** (1971), 127–136.

[9] A.E. Feldmann, J. Könemann, N. Olver, and L. Sanità, *On the equivalence of the bidirected and hypergraphic relaxations for Steiner tree*, Math. Program. **160** (2016), 379–406.

[10] M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl, *Thinning out Steiner trees: A node-based model for uniform edge costs*, Math. Program. Computation **9** (2017), 203–229.

[11] Z.H. Fu and J.K. Hao, *Swap-vertex based neighborhood for Steiner tree problems*, Math. Program. Computation **9** (2017), 297–320.

[12] G. Gamrath, T. Koch, S.J. Maher, D. Rehfeldt, and Y. Shinano, *SCIP-Jack—a solver for STP and variants with parallelization extensions*, Math. Program. Computation **9** (2017), 231–296.

[13] M.X. Goemans, 1996. Unpublished manuscript.

[14] M.X. Goemans and Y.S. Myung, *A catalog of Steiner tree formulations*, Networks **23** (1993), 19–28.

[15] M.X. Goemans, N. Olver, T. Rothvoß, and R. Zenklusen, *Matroids and integrality gaps for hypergraphic Steiner tree relaxations*, Proceedings of the 44th Symposium on Theory of Computing Conference, ACM, 2012, pp. 1161–1176.

[16] L. Gouveia and J. Telhada, *The multi-weighted Steiner tree problem: A reformulation by intersection*, Comput. Oper. Res. **35** (2008), 3599–3611.

[17] Gurobi Optimization, Inc., 2016. Gurobi optimizer reference manual.

[18] S. Hougardy, J. Silvanus, and J. Vygen, *Dijkstra meets Steiner: A fast exact goal-oriented Steiner tree algorithm*, Math. Program. Computation **9** (2017), 135–202.

[19] F.K.H. Hwang, D.S. Richards, and P. Winter, *The Steiner Tree Problem* Vol. 53 of *Annals of Discrete Mathematics*, Elsevier, 1992.

[20] R.M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, Springer US, 1972, pp. 85–103.

[21] T. Koch, Rapid mathematical programming, Ph.D. thesis, Technische Universität Berlin, 2004.

[22] T. Koch, A. Martin, and S. Voß, *SteinLib: An updated library on Steiner tree problems in graphs*, Tech. report ZIB-Report 00-37, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2000.

[23] J. Könemann, D. Pritchard, and K. Tan, *A partition-based relaxation for Steiner trees*, Math. Program. **127** (2011), 345–370.

[24] T. Polzin and S.V. Daneshmand, *A comparison of Steiner tree relaxations*, Discr. Appl. Math. **112** (2001), 241–261.

[25] T. Polzin and S.V. Daneshmand, *On Steiner trees and minimum spanning trees in hypergraphs*, Oper. Res. Lett. **31** (2003), 12–20.

[26] S. Rajagopalan and V.V. Vazirani, *On the bidirected cut relaxation for the metric Steiner tree problem*, Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 1999, pp. 742–751.

[27] G. Robins and A. Zelikovsky, *Tighter bounds for graph Steiner tree approximation*, SIAM J. Discret. Math. **19** (2005), 122–134.

[28] D.M. Warme, Spanning trees in hypergraphs with applications to Steiner trees, Ph.D. thesis, University of Virginia, 1998.

[29] R.T. Wong, *A dual ascent approach for Steiner tree problems on a directed graph*, Math. Program. **28** (1984), 271–287.