Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

Daniel Rehfeldt, Henriette Franz, Thorsten Koch

Optimal Connected Subgraphs: Formulations and Algorithms

ZIB Report 20-23 (August 2020)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Optimal Connected Subgraphs: Formulations and Algorithms

Daniel Rehfeldt, Henriette Franz, Thorsten Koch

Zuse Institute Berlin,

Takustr. 7, 14195 Berlin, Germany

{rehfeldt,koch}@zib.de

TU Berlin, Chair of Software and Algorithms for Discrete Optimization,

Str. des 17. Juni 135, 10623 Berlin, Germany

August 6, 2020

Abstract

Connectivity is a central concept in combinatorial optimization, graph theory, and operations research. In many applications, one is interested in finding an optimal subset of vertices with the essential requirement that the vertices are connected, but not how they are connected. I.e., it is not relevant, which edges are selected to obtain connectivity. Two natural examples in this category are the maximum-weight connected subgraph problem (MWCSP) and the uniform weight (prize-collecting) Steiner tree problem.

This article is concerned with the exact solution of such problems via integer programming. On the theoretical side, we analyze and compare IP and MIP formulations with respect to the strength of their LP relaxations. Along the way, we also provide a tighter (compact) description of the connected subgraph polytope—the convex hull of subsets of vertices that induce a connected subgraph. Furthermore, we give a (compact) complete description of the connected subgraph polytope for graphs with no four independent vertices. On the algorithmic side, we introduce new components, such as primal and dual heuristics, to enhance a branch-and-cut algorithm based on the strongest previously considered IP formulation. These developments allow us to solve MWCSP benchmark instances from the literature faster than current state-of-the-art solvers. Additionally, previously intractable instances can be solved to optimality.

1 Introduction

In many clustering and network analysis applications one is interested in finding an optimal subset of vertices with the main requirement being that the vertices are connected, but not how they are connected. I.e., one looks for a subsets of vertices, such that the subgraph *induced* by these vertices is connected. Which edges are selected to obtain connectivity is not relevant.

Applications of such a *induced connectivity* span a diverse set of areas: Computational biology Dittrich et al. [2008], wildlife conservation Dilkina and Gomes [2010], computer vision Chen and Grauman [2012], social network analysis Moody and White [2003], political districting Garfinkel and Nemhauser [1970], wireless sensor network design Buchanan et al. [2015], and even robotics Banfi [2018].

From an optimization perspective, a fundamental model for such problems is the *maximum-weight connected subgraph problem* (MWCSP), see e.g. Álvarez-Miranda et al. [2013a]. Given an undirected graph $G = (V, E)$ and vertex weights $p : V \to \mathbb{R}$, the task is to find a connected subgraph $S = (V(S), E(S)) \subseteq G$ such that

$$\sum_{v \in V(S)} p(v)$$

is maximized. The literature also describes variations of the MWCSP such as the rooted and the budget constrained problem, see Álvarez-Miranda et al. [2013b]. Another well-known optimization problem that is based on induced connectivity is the unweighted (as well as uniformly weighted) Steiner tree problem: Any solution (i.e. Steiner tree) consisting of n nodes will be of weight $n - 1$; it does not matter which $n - 1$ edges are selected as long as they connect the given nodes. This observation is exploited in Fischetti et al. [2017] for a node based (prize-collecting) Steiner tree algorithm. The corresponding solver won most of the categories at the 11th DIMACS Challenge, dedicated to Steiner tree and related problems. As to the MWCSP, other practical algorithms can for example be found in Álvarez-Miranda et al. [2013a], Leitner et al. [2018], Rehfeldt and Koch [2019].

Various articles discuss theoretical aspects of optimization problems based on induced connectivity, such as the strength of (mixed) integer-programming formulations, e.g. Álvarez-Miranda et al. [2013b], Carvajal et al. [2013], polyhedral descriptions, e.g. Biha et al. [2015], Wang et al. [2017], or complexity, e.g. Álvarez-Miranda et al. [2013a], Buchanan et al. [2018].

This article aims at improving the exact solution of optimization problems based on induced connectivity via integer programming. A central component is an improved theoretical understanding of different IP and MIP formulations, including the relative strength of their LP-relaxations. However, this article also seeks to improve the practical performance of IP algorithms based on these theoretical results.

1.1 Contribution

The first, and main, part of this article analyzes integer and mixed integer formulations for optimization problems that are based on induced connectivity. In particular, node based formulations (which have gained notable attention in the recent literature) are compared with edge based ones. It will be shown that the latter prevail with respect to the strength of their LP-relaxations. Furthermore, polyhedral results are given, including a (compact extended) description of the connected subgraph polytope for all graphs with less than four independent vertices.

The second part of the article introduces algorithmic components to improve the practical exact solution of MWCSP based on the strongest of the previously studied IP formulation. The resulting branch-and-cut solver is shown to be faster than the current state of the art for MWCSP. Furthermore, previously intractable benchmark instances can be solved to optimality.

Both parts of this article are clustered around the MWCSP. However, it will also be shown how to apply the (theoretical and algorithmic) results to other induced connectivity problems, such as (unweighted) Steiner tree.

1.2 Definitions and notation

For the vertices and edges of an undirected, graph G we write $V(G)$ and $E(G)$, respectively. For a directed graph D, we write $A(D)$ for its set of arcs. For a subset of vertices $U \subseteq V$ we define

$$E[U] := \big\{\{v, w\} \in E \mid v, w \in U\big\}.$$

Further, the notation $n := |V|$ and $m := |E|$ will be used.

For $U \subseteq V$ define $\delta(U) := \big\{\{u,v\} \in E \mid u \in U, v \in V \setminus U\big\}$ and for a subgraph $G' \subseteq G$ and $U' \subseteq V(G')$ define $\delta_{G'}(U') := \big\{\{u,v\} \in E(G') \mid u \in U', v \in V(G') \setminus U'\big\}$. A corresponding notation is used for directed graphs (V,A): For $U \subseteq V$ define $\delta^+(U) := \{(u,v) \in A \mid u \in U, v \in V \setminus U\}$ and $\delta^-(U) := \delta^+(V \setminus U)$. For a single vertex v we use the short-hand notation $\delta(v) := \delta(\{v\})$, and accordingly for directed graphs. We define the *neighborhood* of a vertex set $U \subseteq V$ as

$$N(U) := \big\{v \in V \setminus U \mid \exists u \in U, \{u,v\} \in \delta(U)\big\}.$$

For a single $v \in V$ we set $N(v) := N(\{v\})$. For directed graphs we define

$$N^+(U) := \big\{v \in V \setminus U \mid \exists u \in U, (u,v) \in \delta^+(U)\big\}.$$

We denote by $\alpha(G)$ the maximum number of independent vertices in graph G. Given a r-t flow f, we denote its *net flow value* by $v(f) = f(\delta^+(r)) - f(\delta^-(r))$.

Let v and w be two distinct vertices of G. A subset $C \subseteq V \setminus \{v,w\}$ is called $(v,w)-$*separator*, or $(v,w)-$*node-separator*, if there is no path from v to w in the graph $(V \setminus C, E[V \setminus C])$. The family of all $(v,w)-$separators is denoted by $\mathcal{C}(v,w)$. Note that $\mathcal{C}(v,w) = \emptyset$ if and only if $\{v,w\} \in E$. For directed graphs we say that $C \subseteq V \setminus \{v,w\}$ is a $(v,w)-$separator if all directed paths from v to w contain a vertex from C.

For any function $x : M \mapsto \mathbb{R}$ with M finite, and any $M' \subseteq M$ define $x(M') := \sum_{i \in M'} x(i)$. Given an IP formulation F we denote its optimal objective value by $v(F)$. Further, we denote the optimal objective value and the set of feasible points of its LP relaxation by $v_{LP}(F)$ and $\mathcal{P}_{LP}(F)$, respectively. If we want to emphasize a specific problem instance I, we also write $F(I)$.

1.3 Preliminaries: MWCSP and related problems

The MWCSP is $\mathcal{NP}$-hard, see e.g. Johnson [1985]. It is even $\mathcal{NP}$-hard to approximate the MWCSP within any constant factor as shown in Álvarez-Miranda et al. [2013a]. Note that in the case of only non-negative vertex weights, the MWCSP reduces to finding a connected component of maximum vertex weight; in the case of only non-positive vertex weights, the empty set constitutes an optimal solution.

Rooted MWCSP

A close relative of the MWCSP is the *rooted maximum-weight connected subgraph problem* (RMWCSP), see e.g. Álvarez-Miranda et al. [2013b], which incorporates the additional condition that a non-empty set $T_f \subseteq V$ needs to be part of any feasible solution. For simplicity, we usually assume that $p(t) = 0$ for all $t \in T_f$.

Unweighted Steiner tree problem

Given an undirected connected graph $G = (V,E)$ and a set $T \subseteq V$ of *terminals*, the unweighted Steiner tree problem (USPG) is to find a tree $S \subseteq G$ with $T \subseteq V(S)$ such that $|E(S)|$ is minimized. The USPG can also be seen as a Steiner tree problem with uniform edge weights. Moreover, the USPG be formulated as a RMWCSP by assigning each non-terminal vertex a weight of -1. Many of the hardest Steiner tree benchmark instances are unweighted, see Koch et al. [2001] for an overview. Moreover, many theoretical articles consider just the unweighted case, see e.g. Nederlof [2013] for recent complexity results.

Steiner arborescence problem

Several results of this article rely on the *Steiner arborescence problem* (SAP), which is defined as follows: We are given a directed graph $D = (V, A)$, costs $c : A \to \mathbb{R}_{\geq 0}$, a set $T \subseteq V$ of *terminals* and a root $r \in T$. The SAP requires an arborescence (i.e. directed tree) $S \subseteq D$ with $T \subseteq V(S)$ that is rooted at r, such that $c(A(S))$ is minimized.

2 Formulations for rooted connected subgraphs

This section is concerned with connected subgraph problems where a predefinded, non-empty set of vertices needs to be part of any feasible solution. We start with formulations for the SAP. While the SAP is not based on induced connectivity itself, it forms the base of several other results in this article.

2.1 Formulations for the Steiner arborescence problem

Consider an SAP (V, A, T, r, c). Associate with each arc $a \in A$ a binary variable $y(a)$ indicating whether a is contained in the Steiner arborescence ($y(a) = 1$) or not ($y(a) = 0$). A natural formulation by Wong [1984] (i.e., one in the original variable space) can thereupon be stated as:

Formulation 1. *Directed Cut Formulation (DCut)*

$$\begin{aligned} \mathit{min} \quad & c^T y & & (1)\\ \mathit{s.t.}\ y(\delta^-(U)) \quad & \geq \quad 1 & \textit{for all } U \subset V,\ r \notin U, U \cap T \neq \emptyset, \quad & (2)\\ y(a) \quad & \in \quad \{0,1\} & \textit{for all } a \in A. \quad & (3) \end{aligned}$$

Another well-known formulation, see e.g. Wong [1984], is based on flows.

Formulation 2. *Directed Multicommodity Flow Formulation (DF)*

$$\begin{aligned} \mathit{min} \quad & c^T y & & (4)\\ \mathit{s.t.}\ f^t(\delta^-(v)) - f^t(\delta^+(v)) = & \begin{cases} 1 & \textit{if } v = t;\\ 0 & \textit{if } v \in V \setminus \{r, t\} \end{cases} & \textit{for all } v \in V, t \in T \setminus \{r\}, \quad & (5)\\ & f^t \leq y & \textit{for all } t \in T \setminus \{r\}, \quad & (6)\\ & f^t \geq 0 & \textit{for all } t \in T \setminus \{r\}, \quad & (7)\\ & y \in \{0,1\}^A. & & (8) \end{aligned}$$

By using the max-flow min cut theorem, one shows that DF is an extended formulation of $DCut$, i.e., $proj_y(\mathcal{P}_{LP}(DF)) = \mathcal{P}_{LP}(DCut)$, see e.g. Duin [1993]. Both formulations can be strengthened by the so called *flow-balance constraints* from Koch and Martin [1998]:

$$y(\delta^-(v)) \leq y(\delta^+(v)) \qquad \text{for all } v \in V \setminus T. \tag{9}$$

We well refer to the extensive of the above formulations that additionally include (9) as $DCut_{FB}$ and DF_{FB}, respectively.

We end this section with a (new) result for SAP, which will be used several times in the following.

Lemma 1. *If* $|T| \leq 3$, *then* $v_{LP}(DCut_{FB}) = v(DCut_{FB})$.

Proof. For the case of $|T| = 1$ and $|T| = 2$ the lemma holds already without the flow-balance constraints. So let (V, A, T, c, r) be an SAP with two terminals t, u besides the root r. We additionally require that a feasible solution does not have any leaves apart from r, t, u. For this so called two-terminal Steiner tree problem a complete polyhedral description is given by Ball et al. [1989]:

$$f(\delta^-(v)) - f(\delta^+(v)) \geq \begin{cases} -1 & \text{if } v = r; \\ 0 & \text{otherwise;} \end{cases} \quad \text{for all } v \in V, \tag{10}$$

$$(f + f^t)(\delta^-(v)) - (f + f^t)(\delta^+(v)) = \begin{cases} 1 & \text{if } v = t; \\ -1 & \text{if } v = r; \\ 0 & \text{otherwise;} \end{cases} \quad \text{for all } v \in V, \tag{11}$$

$$(f + f^u)(\delta^-(v)) - (f + f^u)(\delta^+(v)) = \begin{cases} 1 & \text{if } v = u; \\ -1 & \text{if } v = r; \\ 0 & \text{otherwise;} \end{cases} \quad \text{for all } v \in V, \tag{12}$$

$$f + f^t + f^u \leq y, \tag{13}$$

$$y, f, f^t, f^u \in \mathbb{R}^A_{\geq 0}. \tag{14}$$

The above description is based on the following observation: Any feasible arborescence for the two-terminal Steiner tree problem consists of a path from r to a splitter node v, as well as a v-t and a v-u path. Note that any of these paths can be a single node.

Let (f^t, f^u, y) be an optimal LP solution to DF_{FB}. Assume that this solution is minimal, i.e. for any feasible solution $(\tilde{f}^t, \tilde{f}^u, \tilde{y}) \leq (f^t, f^u, y)$ it holds that $(\tilde{f}^t, \tilde{f}^u, \tilde{y}) = (f^t, f^u, y)$. We will show that there exist $\hat{f}, \hat{f}^t, \hat{f}^u \in \mathbb{R}^A$ such that $(\hat{f}, \hat{f}^t, \hat{f}^u, y)$ is contained in the polyhedron described above. Define for all $a \in A$:

$$\hat{f}(a) := \min\{f^t(a), f^u(a)\}, \tag{15}$$

$$\hat{f}^t(a) := \max\{f^t(a) - f^u(a), 0\}, \tag{16}$$

$$\hat{f}^u(a) := \max\{f^u(a) - f^t(a), 0\}. \tag{17}$$

First, we show (10). Let $v \in V$. Because of the assumed minimality of (f^t, f^u, y) we obtain:

$$\hat{f}(\delta^-(v)) - \hat{f}(\delta^+(v)) = (f^t + f^u - y)(\delta^-(v)) - (f^t + f^u - y)(\delta^+(v)) \tag{18}$$

$$= (f^t + f^u)(\delta^-(v)) - (f^t + f^u)(\delta^+(v)) + y(\delta^+(v)) - y(\delta^-(v)). \tag{19}$$

If $v = r$, then

$$(f^t + f^u)(\delta^-(v)) - (f^t + f^u)(\delta^+(v)) = -2 \tag{20}$$

and

$$y(\delta^+(v)) - y(\delta^-(v)) \geq 1, \tag{21}$$

thus (19) implies that (10) holds. If $v \in \{t, u\}$, then

$$(f^t + f^u)(\delta^-(v)) - (f^t + f^u)(\delta^+(v)) = 1 \tag{22}$$

and

$$y(\delta^+(v)) - y(\delta^-(v)) \geq -1. \tag{23}$$

Finally, if $v \in V \setminus \{r, t, u\}$, the flow-balance constraints imply that (19) is non-negative.

Next, consider (11)—and equivalently (12). By definition it holds that

$$(\hat{f} + \hat{f}^t)(\delta^-(v)) - (\hat{f} + \hat{f}^t)(\delta^+(v)) = f^t(\delta^-(v)) - f^t(\delta^+(v)), \tag{24}$$

which implies (11). Likewise, (13) follows from the definition of $\hat{f}, \hat{f}^t$, and $\hat{f}^u$. □

Note that the lemma is best possible in the sense that there exist SAP instances with $|T| = 4$ such that $v_{LP}(DCut_{FB}) \neq v(DCut_{FB})$, see e.g. Liu [1990], Polzin and Daneshmand [2001a].

2.2 Rooted maximum-weight connected subgraphs

This section discusses the directed variant of the RMWCSP, see Álvarez-Miranda et al. [2013b]: Given a directed graph $D = (V, A)$, vertex weights $p : V \to \mathbb{R}$, a non-empty set $T_f \subseteq V$ and an $r \in T_f$, find a connected subgraph $S \subseteq D$ containing R such that any $v \in V(S)$ can be reached from r on a directed path in S, and $p(V(S))$ is maximized. Any undirected RMWCSP can be formulated in directed form by choosing an arbitrary $r \in T_f$ and replacing each edge by two anti-parallel arcs.

Note that any solution to the directed RMWCSP can be represented as an arborescence. This observation leads to the following IP formulation, see e.g. Álvarez-Miranda et al. [2013b], based on a well-known formulation for SAP, see e.g. Goemans and Myung [1993]. Define for each $v \in V$ a variable $x(v) \in \{0, 1\}$ that is equal to 1 if and only if vertex v is part of the solution. Analogously, define for each $a \in A$ a variable $y(a) \in \{0, 1\}$.

Formulation 3. *Rooted Steiner Arborescence Formulation (RSA)*

$$
\begin{aligned}
&max\ p^T x && && (25)\\
&s.t.\ y(\delta^-(v)) = x(v) && \text{for all } v \in V \setminus \{r\} && (26)\\
&\quad y(\delta^-(U)) \geq x(v) && \text{for all } U \subseteq V \setminus \{r\},\ v \in U && (27)\\
&\quad x(t) = 1 && \text{for all } t \in T_f && (28)\\
&\quad x \in \{0,1\}^V && && (29)\\
&\quad y \in \{0,1\}^A. && && (30)
\end{aligned}
$$

In Álvarez-Miranda et al. [2013b] a new formulation for the directed RPCSTP based on node-separators is introduced. Note that the use of node-separators for modeling connectivity is already suggested in Fügenschuh and Fügenschuh [2008].

Formulation 4. *Rooted Node Separator Formulation (RNCut)*

$$
\begin{aligned}
max \quad & p^T x && && (31)\\
s.t.\ x(C) \quad & \geq \quad x(v) && \text{for all } v \in V \setminus (\{r\} \cup N^+(r)),\ C \in \mathcal{C}(r, v) && (32)\\
x(v) \quad & = \quad 1 && \text{for all } v \in T_f && (33)\\
x \quad & \in \quad \{0,1\}^A. && && (34)
\end{aligned}
$$

Besides the two IP models introduced above, several other formulations for RMWCSP (sometimes including a budget constraint) have been introduced in the literature, see e.g. Álvarez-Miranda et al. [2013b], Dilkina and Gomes [2010]. However, one can show that these formulations are weaker with respect to the LP relaxation than both of the above models, see Álvarez-Miranda et al. [2013b] for some such results. Another example is the formulation from Conrad et al. [2007] that is based on single-flow. However, also this formulation can be shown to be weaker than Formulation 3 by using max-flow/min-cut arguments—similarly to corresponding results for minimum spanning tree or Steiner tree problems, which can be found for example in Hwang et al. [1992].

In Álvarez-Miranda et al. [2013b] it is stated that the LP relaxations of the RNCut and RSA model yield the same optimal value. Unfortunately, this claim is not correct, as the following proposition shows. Appendix A discusses the error in the line of argumentation in Álvarez-Miranda et al. [2013b]—and furthermore provides some insight on how the node separator constraints miss to capture structures accurately described by edge cut constraints.

Proposition 2. *It holds that* $proj_x(\mathcal{P}_{LP}(RSA)) \subset \mathcal{P}_{LP}(RNCut)$ *and the inclusion can be strict.*

Proof. The inclusion is essentially proven in Álvarez-Miranda et al. [2013b]. An example for a strict inclusion is given in Appendix A. □

One can strengthen the RSA formulation by inequalities similar to the flow-balance constraints. However, these constraints depend on the objective vector, so they cannot (directly) be used for polyhedral results.

$$y(\delta^-(v)) \leq y(\delta^+(v)) \qquad \text{for all } v \in V \setminus (T_f \cup T_p). \tag{35}$$

We refer to the strengthened formulation as RSA_{FB}. One readily obtains the following result from Lemma 1.

Lemma 3. *If* $|T_p \cup T_f| \leq 3$, *then* $v_{LP}(RSA_{FB}) = v(RSA_{FB})$.

Proof. Let I be an RMWCSP instance with $|T_p \cup T_f| \leq 3$. Define an SAP $I' = (V', A', T', c')$ on an extended graph (V', A') as described in Rehfeldt and Koch [2019] for the case of $|T_f| = 1$. Initially, set $V' := V$, $A' := A$, and $T' := T_f$. For each arc $a = (v, w) \in A$ set $c'(a) := \max\{-p(w), 0\}$. For each $t \in T_p$ we add a new terminal t' to T' and arcs (r, t') of weight $p(t)$ and (t, t') of weight 0 to A'. It holds that

$$p(T_p) - v(DCut_{FB}(I')) = v(RSA_{FB}(I)); \tag{36}$$

recall that we assume $T_p \cap T_f = \emptyset$. Any optimal LP solution (y, z) to RSA_{FB} can be extended to a feasible LP solution y' to $DCut_{FB}$ defined by $y'(t, t') = x(t)$, $y'(r, t') = 1 - x(t)$ for all $t \in T_p$, as well as $y'(a) := y(a)$ for all $a \in A$. Thus,

$$p(T_p) - v_{LP}(DCut_{FB}(I')) \geq v_{LP}(RSA_{FB}(I)) \geq v(RSA_{FB}(I)). \tag{37}$$

Because I' has at most three terminals, Lemma 1 and (36) imply that the above inequalities are satisfied with equality. Consequently, $v_{LP}(RSA_{FB}(I)) = v(RSA_{FB}(I))$. □

2.3 Unweighted Steiner tree problems

This section analyzes and compares two formulations for the USPG. First, we state the node-separator formulation from Fischetti et al. [2017]. Note that in Fischetti et al. [2017] a more general version for the prize-collecting USPG is used. However, the prize-collecting USPG is essentially a MWCSP. The results of this section can be partly extended to this more general variant (which is done in Section 3 for the non-rooted case), but for simplicity, we now consider the USPG only.

Formulation 5. *Terminal Node Separator Formulation (TNCut)*

$$\begin{aligned}
\text{min} \quad & x(V) - 1 && && (38)\\
\text{s.t.} \quad x(C) &\geq 1 && \text{for all } t,u \in T, t \neq u, C \in \mathcal{C}(t,u), && (39)\\
x(v) &= 1 && \text{for all } v \in T, && (40)\\
x(v) &\in \{0,1\} && \text{for all } v \in V. && (41)
\end{aligned}$$

Second, we look at the well-known *bidirected cut formulation* (*BDCut*) for (U)SPG. This formulation corresponds to the *DCut* formulation for the SAP obtained by replacing each edge of the SPG by two anti-parallel arcs of the same weight, and choosing an arbitrary terminal as the root.

2.3.1 Exactness of the bidirected cut formulation

This section formulates conditions under which the bidirected cut formulation has no integrality gap. We start with a direct consequence of Lemma 1, which applies also to weighted SPG.

Proposition 4. *If* $|T| \leq 3$*, then* $v_{LP}(BDCut_{FB}) = v(BDCut_{FB})$*.*

A simple reduction technique for USPG is to contract adjacent terminals (and delete one edge from each resulting pair of multi-edges). The following proposition shows that the absolute integrality gap of *BDCut* is invariant under this operation.

Proposition 5. *Let* I *be an USPG instance with adjacent terminals* t, u*. Let* I' *be the USPG obtained from contracting* t *and* u*. It holds that:*

$$v_{LP}(BDCut(I)) = v_{LP}(BDCut(I')) + 1. \qquad (42)$$

Proof. Throughout the proof we assume that u is the root for the *BDCut* formulation, i.e. $r = u$. It is well-known that the choice of the root does not affect $v_{LP}(BDCut)$ (this result also follows from the proof of Theorem 6). Furthermore, let $D' = (V', A')$ be the bidirected graph obtained by contracting r and t and let r' be the new vertex. I.e., $V' = (V \setminus \{r,t\}) \cup \{r'\}$.

First, we show that $v_{LP}(BDCut(I)) \geq v_{LP}(BDCut(I'))+1$. Let y be an optimal LP solution to $BDCut(I)$. The optimality of y implies that $y(\delta^-(t)) = 1$, see Polzin and Daneshmand [2001a]. Create a new optimal solution $\tilde{y}$ as follows. Set $\tilde{y}(a) := y(a)$ for all $a \in A \setminus \delta^-(t)$, $\tilde{y}(a) := 0$ for all $a \in \delta^-(t) \setminus \{(r,t)\}$, and $\tilde{y}((r,t)) := 1$. Note that for any cut $\delta^-(U)$ with $U \subset V \setminus \{r\}$ such that $\delta^-(U) \cap \delta^-(t) \neq \emptyset$ it holds that $(r,t) \in \delta^-(U)$. Thus, $\tilde{y}(\delta^-(U)) \geq 1$. Consequently, $\tilde{y}$ satisfies (2). Define an LP solution y' to $BDCut(I')$ as follows: $y'(a) := \tilde{y}(a)$ for all $a \in A' \cap A$, and $y'(a) := 0$ for all $a \in \delta^-_{D'}(r')$. For any $a = (r', v) \in \delta^+_{D'}(r')$ proceed as follows. If $(r,v),(t,v) \in A$, set $y'(a) := \tilde{y}((r,v)) + \tilde{y}((t,v))$; if $(t,v) \notin A$, set $y'(a) := \tilde{y}((r,v))$; otherwise, set $y'(a) := \tilde{y}((t,v))$. Because of $y(\delta^-(v)) \leq 1$, we have in any case that $y'(a) \leq 1$.

It remains to show that $v_{LP}(BDCut(I)) \leq v_{LP}(BDCut(I'))+1$. Given an optimal LP solution y' to $BDCut(I')$ we define a corresponding LP solution y to $BDCut(I)$. First, $y((r,t)) := 1$, $y((t,r)) := 0$. Second, $y(a) := y'(a)$ for all $a \in A' \cap A$, and $y(a) := 0$ for all $a \in \delta^-(\{r,t\})$. Next, consider the remaining edges $\delta^+(\{r,t\})$. If $(r,v),(t,v) \in A$ set $y((r,v)) := y'(r',v)$, $y((t,v)) := 0$; otherwise, for $a = (r,v)$ or $a = (t,v)$ set $y(a) = y'((r',v))$. □

With this result at hand, we obtain the following theorem (recall that $\alpha(G)$ denotes the independence number of graph G).

Theorem 6. *Consider an USPG on a graph* G*. If* $\alpha(G) \leq 3$*, then* $v_{LP}(BDCut) = v(BDCut)$*.*

Proof. Consider a USPG instance $I = (G, T, c)$ with $\alpha(G) \leq 3$. Let $I' = (G', T', c')$ be the USPG obtained by (repeatedly) contracting all adjacent terminals. Let $D' = (V', A')$ be the bidirected equivalent of G'. Proposition 5 implies that $v_{LP}(BDCut(I)) = v(BDCut(I))$ if and only if $v_{LP}(BDCut(I')) = v(BDCut(I'))$. Furthermore, because of $\alpha(G) \leq 3$ it holds that $|T'| \leq 3$. For $|T'| < 3$, the $BDCut$ formulation is well-known to have no integrality gap. So assume $|T'| = 3$. By construction of I', the terminals form an independent set. Further, let y be an optimal LP solution to $BDCut(I')$ with an arbitrary $r \in T'$ being the root.

Suppose that $v_{LP}(BDCut(I')) \neq v(BDCut(I'))$. By Lemma 1, there is a $v \in V' \setminus T'$ such that

$$y(\delta^+(v)) < y(\delta^-(v)). \tag{43}$$

Because of $\alpha(G) \leq 3$, at least one of the terminals needs to be adjacent to v. We may assume that this property holds for r. Otherwise, we can readily create another optimal LP solution $\tilde{y}$ that satisfies (43) and has a root adjacent to v: Assume that a $t \in T \setminus \{r\}$ is adjacent to v and let f^t be a unit flow from r to t such that $f^t \leq y$; define $\tilde{y}((q, u)) := y((q, u)) - f^t((q, u)) + f^t((u, q))$ for all $(u, q) \in A'$.

Define a new LP solution y' from y as follows. For $a_0 := (r, v)$ set $y'(a_0) := y(\delta^+(v))$. For any $a \in \delta^-(v) \setminus \{a_0\}$ set $y'(a) := 0$. For all (remaining) $a \in A' \setminus \delta^-(v)$ set $y'(a) := y(a)$. Note that because of (43) it holds that $y'(A') < y(A')$. It remains to be shown that y' is feasible. Suppose that there is a $U \subseteq V \setminus \{r\}$ with $U \cap T' \neq \emptyset$ and $y'(\delta^-(U)) < 1$. Because y is feasible, it has to hold that $v \in U$. Let $\tilde{U} := U \setminus \{v\}$. By the construction of y' it holds that

$$\begin{aligned} y(\delta^-(\tilde{U})) &= y'(\delta^-(\tilde{U})) \\ &= y'(\delta^-(\tilde{U})) + y'((r, v)) - y'(\delta^+(v)) \\ &\leq y'(\delta^-(U)) \\ &< 1, \end{aligned}$$

which contradicts the feasibility of y. Consequently, we have shown that $v_{LP}(BDCut(I')) = v(BDCut(I'))$ and, thus, $v_{LP}(BDCut(I)) = v(BDCut(I))$. □

The theorem is best possible; i.e., there exist USPG instances such that $\alpha(G) = 4$ and $v_{LP}(BDCut) \neq v(BDCut)$, see e.g. Duin [1993], Filipecki and Van Vyve [2020].

2.3.2 Comparison of edge and node based formulation

Formulation 5 ($TNCut$) was used within a branch-and-cut algorithm by the most successful solver at the 11th DIMACS Challenge DIMACS. Furthermore, the solver was able to solve several USPG benchmark instances that had been unsolved for more than a decade to optimality. Thus, one might wonder how this formulation theoretically compares with the better known bidirected cut formulation. As the next proposition shows, $BDCut$ is always stronger than $TNCut$ and the relative gap can be rather large.

Proposition 7. *It holds that $v_{LP}(TNCut) \leq v_{LP}(BDCut)$. Furthermore,*

$$\sup \left\{ \frac{v_{LP}(BDCut)}{v_{LP}(TNCut)} \right\} \geq 2, \tag{44}$$

where the supremum is taken over all USPG instances.

Proof. For the first inequality consider an optimal LP solution y to $BDCut$. Define $x \in \mathbb{R}^V$ by $x(v) := y(\delta^-(v))$ for all $v \in V \setminus \{r\}$ and $x(r) := 1$. The optimality of y implies $x(v) \leq 1$ for all v, see Polzin and Daneshmand [2001a]. Let $t, u \in T$ with $t \neq u$ and $C_{tu} \in \mathcal{C}(t,u)$. We will show that $C(t,u)$ satisfies (39). If $C_{tu} \cap T \neq \emptyset$, then $x(C_{tu}) \geq 1$, because $x(q) \geq 1$ for all $q \in T$ due to (2) and the definition of x. Thus, (39) holds. If $C_{tu} \cap T = \emptyset$, let U_r be the connected component in the graph induced by $V \setminus C_{tu}$ with $r \in U_r$. By definition of C_{tu}, either $t \notin U_r$ or $u \notin U_r$. Therefore, $y(\delta^+(U_r)) \geq 1$, which implies $y(\delta^-(C_{tu})) \geq 1$ because of $\delta^+(U_r) \subset \delta^-(C_{tu})$. Now we obtain from the definition of x that

$$x(C_{tu}) \geq y(\delta^-(C_{tu})) \geq 1.$$

Finally, by construction of x we have that

$$x(V) - 1 = \sum_{v \in V} y(\delta^-(v)) = y(A);$$

note that $y(\delta^-(r)) = 0$ because y is optimal.

For (44) we construct the following family of USPG instances. For any $k \geq 3$ let I_k be the USPG instance with with $k + k^2$ nodes, $k + k^2$ edges, and k terminals defined as follows. Let t_i for $i = 1, ...k$ be the terminals and define for each $i \in \{1, ..., k\}$ Steiner nodes $v_{i,j}$, $j = 1, ...k$. For each $i \in \{1, ..., k\}$ define edges $\{t_i, v_{i,1}\}$, $\{t_{(i+1) \mod k}, v_{i,k}\}$, and $\{v_{i,j}, v_{i,j+1}\}$ for $j = 1, ..., k-1$. Instance I_3 is shown in Figure 1. A feasible (and indeed optimal) LP solution x to $TNCut(I_k)$ is given by $x(t) := 1$ for all terminals t and $x(v) := 0.5$ for any Steiner node v. Its objective is $\frac{k^2}{2} + k - 1$. On the other hand, $v_{LP}(BDCut(I_k)) = k + k(k-1) = k^2$. Thus,

$$\lim_{k \to \infty} \frac{v_{LP}(DCut(I_k))}{v_{LP}(TNCut(I_k))} = \lim_{k \to \infty} \frac{k^2}{\frac{k^2}{2} + k - 1} = 2, \tag{45}$$

which concludes the proof. □

Corollary 8. *The (relative) integrality gap of $TNCut$ is at least* 2.

Note that one can strengthen $TNCut$ by constraints that correspond to the flow-balance constraints for $BDCut$, see Fischetti et al. [2017]. However, if compared to $BDCut_{FB}$, the results of Proposition 7 remain the same for this stronger version of $TNCut$.

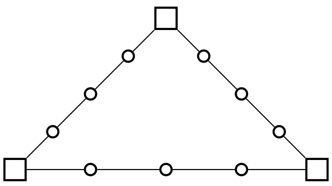

Figure 1: USPG instance I_3. Terminals are drawn as squares.

3 Formulations for non-rooted connected subgraphs

In this section we consider the undirected MWCSP. Some of the following results can also be extended to the directed case. However, the undirected MWCSP is the more common (and, arguably, also more natural) problem.

3.1 Node based formulations

This section considers formulations for MWCSP that use only node variables. The probably best known one, see e.g. Wang et al. [2017], is given below.

Formulation 6. *Node Separator Formulation (NCut)*

$$
\begin{aligned}
& max \quad p^T x & & & (46)\\
s.t.\ x(v) + v(w) - x(C) &\leq 1 & & \text{for all } v, w \in V, v \neq w,\ C \in \mathcal{C}(v, w), & (47)\\
x(v) &\in \{0, 1\} & & \text{for all } v \in V. & (48)
\end{aligned}
$$

The contraction of neighboring positive weight vertices drastically reduces the size of many real-world MWCSP instances, as for example shown in Rehfeldt et al. [2019]. Recalling the invariance of the *BDCut* integrality gap to the contraction of terminals, one might wonder whether a corresponding property holds for MWCSP and the *NCut* formulation. The answer is given in the next proposition. Note that when contracting adjacent vertices $t, u \in T_p$ into a new vertex t', we set $p(t') := p(t) + p(u)$.

Proposition 9. *$v_{LP}(NCut)$ is invariant under the contraction of adjacent vertices of positive weight.*

Proof. Let I be an MWCSP instance with an edge $\{t, u\} \in E$ such that $t, u \in T_p$. Let $I' = (V', E', p')$ be the instance obtained from I be contracting $\{t, u\}$ into a new vertex t'. It holds that $v_{LP}(NCut(I')) \leq v_{LP}(NCut(I))$, because any $x' \in \mathcal{P}_{LP}(NCut(I'))$ can be mapped to a $x \in \mathcal{P}_{LP}(NCut(I))$ with $p^T x = {p'}^T x'$ defined by $x(v) := x'(v)$ for all $v \in V \cap V'$, and $x(t) := x(u) := x'(t')$. The opposite case is somewhat more involved.

Let x be an optimal LP solution to $NCut(I)$. The optimality of x, and the fact that $\{t, u\} \in E$ imply

$$x(t) = x(u). \tag{49}$$

Define $x' \in \mathbb{R}^{V'}$ by $x'(v) := x(v)$ for all $v \in V' \setminus \{t'\}$, and $x'(t') := x(t)$. Assume that $x'(t') \in (0, 1)$—otherwise, the proof is already complete. It remains to be shown that $x' \in \mathcal{P}_{LP}(NCut(I'))$. Suppose this is not the case. Then there are $a, b \in V'$ and an a-b separator $C'_{ab} \subset V'$ such that

$$x'(a) + x'(b) - x'(C'_{ab}) > 1. \tag{50}$$

Because x is feasible, $t' \in C'_{ab}$. Thus, we obtain from (50) that

$$x(a) + x(b) - x(t) = x'(a) + x'(b) - x'(t') > 1, \tag{51}$$

and therefore

$$\min\{x(a), x(b)\} > x(t). \tag{52}$$

Now we return to the original instance I. Because x is optimal, and $x(t) = x(u) < 1$, there is a $q \in V \setminus \{t, u\}$ and a $C_{qt} \in \mathcal{C}(q, t)$ such that

$$x(t) + x(q) - x(C_{qt}) = 1. \tag{53}$$

Similarly, there is a $s \in V \setminus \{t, u\}$ and a $C_{su} \in \mathcal{C}(s, u)$ with $x(u) + x(s) - x(C_{su}) = 1$. At least one such combination q, C_{qt}, or s, C_{su} satisfies $u \notin C_{qt}$ or $t \notin C_{su}$, otherwise we could increase $x(u)$ and $x(t)$. Assume w.l.o.g. $u \notin C_{qt}$. Further, observe that (53) implies

$$x(C_{qt}) \leq \min\{x(t), x(q)\}. \tag{54}$$

Thus, (53) and (52) imply $a, b \notin C_{qt}$. One notes that $C_{qt} \notin \mathcal{C}(a, q)$, because (52) and (53) imply

$$x(a) + x(q) - x(C_{qt}) > 1. \tag{55}$$

Likewise, $C_{qt} \notin \mathcal{C}(b, q)$. Consequently, any path from $\{t, u\}$ to a or b needs to cross C_{qt}; otherwise, the latter would not separate q and t. Therefore, $\tilde{C}_{ab} := (C'_{ab} \setminus \{t'\}) \cup C_{qt}$ separates a and b (in the original graph). However, from (50) and (54) we obtain

$$1 < x(a) + x(b) - x'(C'_{ab}) \leq x(a) + x(b) - x(\tilde{C}_{ab}), \tag{56}$$

which contradicts the feasibility of x. □

Furthermore, one obtains the following optimality criterion:

Proposition 10. *If $|T_p| \leq 2$, then $v_{LP}(NCut) = v(NCut)$.*

Proof. Consider an MWCSP $I = (G, p)$ with $|T_p| \leq 2$. The case $|T_p| \leq 1$ is clear. Let $\{a, b\} := T_p$ and assume $p(a) \geq p(b)$. Thus, there is a minimal optimal LP solution x such that $x(a) = 1$. Let (V, A) be the bidirected equivalent of G. Create a new directed graph (V', A') by replacing each node $v \in V \setminus \{a, b\}$ by two nodes v_1, v_2 and arcs $(v_1, v_2), (v_2, v_1)$. Further, all ingoing arcs of v become ingoing arcs of v_1, and all outgoing arcs of v are now outgoing arcs of v_2. Define arc capacities k for each pair of these new arcs by $x(v)$; for any (remaining) arc $e \in A$ set $k(e) := \infty$.[1] By the max-flow/min-cut theorem there is a-b flow f with $v(f) = x(b)$ in this extended network. Define the directed MWCSP $I_r := ((V, A), T_f, r, p)$ with $T_f := \{a\}$ and $r := a$, and set $y := f \restriction_A$. Because of the optimality and minimality of x it holds that $(x, y) \in \mathcal{P}_{LP}(RSA(I_r))$. Thus, $v_{LP}(NCut(I)) \leq v_{LP}(RSA(I_r))$. Furthermore, y satisfies constraints (35). Because of $v(NCut(I)) = v(RSA(I_r))$, Lemma 3 implies that $v_{LP}(NCut(I)) = v(NCut(I))$. □

Figure 2 shows an MWCSP instance with $|T_p| = 3$ and $v_{LP}(NCut) \neq v(NCut)$. It holds that $v(NCut) = 1$, but $v_{LP}(NCut) = 1.5$ (set the values of all negative weight node variables to 0.5 and the remainder to 1).

Finally, by combining the previous two propositions we obtain a significantly shorter proof of a main result from Wang et al. [2017].

Theorem 11. *If $\alpha(G) \leq 2$, then $\mathcal{P}_{LP}(NCut)$ is integral.*

Proof. Let $p \in \mathbb{R}^V$. If $\alpha(G) \leq 2$, then Proposition 9 implies that the MWCSP (G, p) can be transformed to an MWCSP with at most two positive weight vertices without changing $v_{LP}(NCut)$. Now, Proposition 10 gives $v_{LP}(NCut) = v(NCut)$. Because p can be chosen arbitrarily, $\mathcal{P}_{LP}(NCut)$ is integral. □

Wang et al. [2017] also show that $\mathcal{P}_{LP}(NCut)$ is integral only if $\alpha(G) \leq 2$.

Indegree constraints

Given an undirected graph $G = (V, E)$, a $d \in \mathbb{Z}^n$ is an *indegree vector* if there is an orientation $D = (V, A)$ of G such that $d_v = |\delta^-_D(v)|$ for all $v \in V$. For each indegree vector d the corresponding *indegree inequality* is given as

$$\sum_{v \in V} (1 - d_v) x(v) \leq 1, \tag{57}$$

where $x \in \mathbb{R}^V_{\geq 0}$ are the node variables. Korte et al. [2012] show that the indegree inequalities describe the connected subgraph polytope if G is a tree. Furthermore, Wang et al. [2017] show conditions for (57) to be facet inducing and shows that the constraints can be separated in linear time. It is further shown that the constraints (57) can strengthen the $NCut$ formulation.

[1] Such kind of flow network transformations are well-known in algorithmic graph theory; see also Appendix A.

3.2 Edge based formulations

An edge based formulation for the directed MWCSP is introduced in Álvarez-Miranda et al. [2013a], based on a transformation to the prize-collecting SPG. We will use essentially the same formulation for the undirected MWCSP, but without the transformation to the prize-collecting SPG, and thus with a different objective function. Consider the bidirected equivalent $D = (V, A)$ to the given undirected graph. Let (V_r, A_r) be the directed graph defined as follows with an additional node r: $V_r := V \cup \{r\}$, $A_r := A \cup \{(r, v) \mid v \in V\}$. Define the following extended MWCSP formulation based on the new graph (V_r, A_r).

Formulation 7. *Extended Steiner Arborescence Formulation (ESA)*

$$\max\ p^T x \tag{58}$$

$$\text{s.t. } y(\delta^-(v)) = x(v) \qquad \text{for all } v \in V \tag{59}$$

$$y(\delta^-(U)) \geq x(v) \qquad \text{for all } U \subseteq V,\ v \in U \tag{60}$$

$$y(\delta^+(r)) \leq 1 \tag{61}$$

$$x \in \{0,1\}^V \tag{62}$$

$$y \in \{0,1\}^{A_r}. \tag{63}$$

The remainder this section aims to prove an integrality condition for $proj_x(\mathcal{P}_{LP}(ESA))$ based on the independence number. We follow the same principal ideas used in Section 2.3.1, but require additional technical results, due to the absence of a natural root node. We start with the easiest one.

Lemma 12. *Let (x, y) be an optimal LP solution (x, y) to ESA, and let $v \in V$. There is a $\tilde{y} \in \mathbb{R}^{A_r}$ with $\tilde{y}((r, v)) = x(v)$ such that $(x, \tilde{y})$ is an optimal LP solution to ESA.*

Proof. Assume there is an optimal LP solution (x, y) with $\kappa := y(\delta^-_D(v)) > 0$ for a $v \in V$. Because of (60), there is a r-v flow $f_\kappa \leq y$ with $v(f_\kappa) = \kappa$ and $f_\kappa((r, v)) = 0$. Define a new solution $(x, \tilde{y})$ with

$$\tilde{y}((u,w)) := \begin{cases} y((u,w)) - f_\kappa((u,w)) + f_\kappa((w,u)), & (u,w) \in A \\ y((u,w)) - f_\kappa((u,w)), & (u,w) \in \delta^+(r) \setminus \{(r,v)\} \\ y((u,w)) + \kappa, & (u,w) = (r,v). \end{cases}$$

For (59), first let $u \in V \setminus \{v\}$. It holds that

$$\tilde{y}(\delta^-(u)) = y(\delta^-(u)) + f_\kappa(\delta^+_D(u)) - f_\kappa(\delta^-_D(u)) - f_\kappa((r,u)) = y(\delta^-(u)) = x(u).$$

Similarly, because of $\tilde{y}((r, v)) = y((r, v)) + \kappa$ and $f_\kappa(\delta^-_D(v)) + f_\kappa(\delta^+_D(v)) = \kappa$ it holds that $\tilde{y}(\delta^-(v)) = x(v)$.

For (60), consider a $U \subseteq V$, and a $u \in U$. First, assume $v \in U$. Because of $(r, v) \in \delta^-(U)$, and $f_\kappa(\delta^-_D(U)) + f_\kappa(\delta^+_D(U)) = \kappa$ we obtain $\tilde{y}(\delta^-(U)) = y(\delta^-(U))$. Second, assume $v \notin U$. In this case, flow conservation of f_κ implies that

$$\tilde{y}(\delta^-(U)) = y(\delta^-(U)) + f_\kappa(\delta^+(U)) - f_\kappa(\delta^-(U)) = y(\delta^-(U)).$$

□

Note that for (exact) optimization, ESA only requires arcs from r to vertices in T_p. Furthermore, only constraints (60) for vertices $v \in U$ with $v \in T_p$ need to be enforced. We will refer to this modified formulation as ESA^+. Further, we define $A_r^+ := A \cup \{(r, t) \mid t \in T_p\}$. In practice, it is advisable to add additional $|T_p|$ symmetry breaking constraints similar to those from Fischetti et al. [2017] to ESA^+. As to the LP relaxation of ESA^+, one obtains the following result.

Lemma 13. *Let (x, y^+) be an optimal LP solution to ESA^+. Then $(x, y) \in \mathbb{R}^{V+A_r}$ with $y(a) := y^+(a)$ for $a \in A_r^+$ and $y(a) := 0$ for $a \in A_r \setminus A_r^+$ is an optimal LP solution to ESA.*

Proof. Let ESA' be the reduced version of ESA where constraints (60) are only enforced for vertices $v \in U$ with $v \in T_p$. Note that

$$v_{LP}(ESA) \leq v_{LP}(ESA') \leq v_{LP}(ESA^+). \tag{64}$$

In this proof we only consider minimal optimal LP solutions, i.e., solutions for which no entry can be reduced without loosing either feasibility or optimality.

First, we show that any optimal LP solution to ESA^+ is also optimal for ESA'. To this end, we show the existence of an optimal LP solution (x', y') to ESA' such that $y'((r, v)) = 0$ for all $v \in V \setminus T_p$. Assume there is an optimal LP solution (x', y') to ESA' with $y'((r, v)) > 0$ for a $v \in V \setminus T_p$. Because (x', y') is optimal, there is a r-t flow f^t with $f^t \leq y'$ for a $t \in T_p$ with $v(f^t) = y'((r, v))$. We can now proceed as in Lemma 12 to revert the flow going to t. The resulting optimal solution $(\tilde{x}, \tilde{y})$ satisfies $\tilde{y}((r, v)) = 0$ and $\tilde{y}((r, u)) \leq y'((r, u))$ for all $u \in V \setminus \{t\}$.

Second, we show that any optimal LP solution (x', y') to ESA' with $y'((r, v)) = 0$ for all $v \in V \setminus T_p$ satisfies constraints (60) also for $v \in U$ with $v \notin T_p$. We use essentially the same line of argumentation used in Goemans and Myung [1993] for the SPG bidirected cut formulation. Suppose there is a $U \subseteq V$ and a $u \in U$ with

$$x'(u) > y'(\delta^-(U)). \tag{65}$$

Choose such a U with $|U|$ as small as possible. Because of (65), there is a $e \in \delta^-(u) \setminus \delta^-(U)$ such that $y'(e) > 0$. Because of the minimality of (x', y'), there is a $W \subseteq V$ and a $t \in W \cap T_p$ such that $e \in \delta^-(W)$ and

$$y'(\delta^-(W)) = x'(t). \tag{66}$$

Because of $e \subseteq U$ and $|e \cap W| = 1$, one obtains $|U \cap W| < |U|$. We will show that $U \cap W$ satisfies (65), which contradicts the minimality of $|U|$. By standard graph theory we have that

$$y'((\delta^-(U)) + y'(\delta^-(W)) \geq y'(\delta^-(U \cap W)) + y'(\delta^-(U \cup W)).$$

Together with (66), it follows that $y'((\delta^-(U)) \geq y'(\delta^-(U \cup W))$, which leads to the sought for contradiction. □

Corollary 14. $v_{LP}(ESA) = v_{LP}(ESA^+)$.

Further, we require the following result.

Lemma 15. *If $|T_p| \leq 3$, then there is an optimal LP solution (x, y) to ESA such that $x(t) \in \{0, 1\}$ for all $t \in T_p$.*

Proof. As before, let $D = (V, A)$ be the bidirected equivalent to the given undirected graph. Also, we assume any optimal solution to be minimal. By Lemma 13 we can consider ESA^+ instead of ESA to show the required result. Thus, throughout this proof we consider an optimal LP solution (x, y) to ESA^+.

The case $|T_p| \leq 1$ is clear. Assume $|T_p| = 2$, and let $\{a, b\} := T_p$ such that $p(a) \geq p(b)$. By Lemma 12 we can assume that $y(\delta_D^-(a)) = 0$. Thus, also $y(\delta_D^+(b)) = 0$. If $y(\delta^+(a)) = 0$, either $x(a) = 1$ and $x(b) = 0$, or vice versa. If $y(\delta^+(a)) > 0$, the minimality of (x, y) implies

$$\sum_{v \in V \setminus \{a\}} p(v) y(\delta_D^-(v)) > 0, \tag{67}$$

which implies also $\beta := y(\delta_D^-(b)) > 0$. Let $\kappa := \frac{1}{\beta}$. Define $\tilde{y} \in \mathbb{R}^{A_r^+}$ by $\tilde{y}((r,a)) := 1$, $\tilde{y}((r,b)) := 0$, and $\tilde{y}(e) := \kappa y(e)$ for all $e \in A$. Define $\tilde{x}(v) := \tilde{y}(\delta^-(v))$ for all $v \in V$. One notes that $(\tilde{x}, \tilde{y})$ is feasible, and satisfies $\tilde{x}(a) = \tilde{x}(b) = 1$. Furthermore, $p^T \tilde{x} \geq p^T x$ because of $\kappa \geq 1$.

In the remainder of this proof we consider an MWCSP instance I with $|T_p| = 3$.

Claim 1. *There is an optimal LP solution (x, y) to $ESA^+(I)$ such that $x(t) = 1$ for a $t \in T_p$.*

Proof. Let $\{a, b, c\} := T_p$ such that $p(a) \geq \max\{p(b), p(c)\}$. Again, assume $y((r,a)) = x(a)$. Thus, also $y(\delta_D^-(a)) = 0$. Suppose that $y((r,t)) < 1$ for all $t \in T_p$. Note that $y((r,b)) > 0$ or $y((r,c)) > 0$ (otherwise $y((r,a)) = 1$). Assume w.l.o.g. $y((r,b)) > 0$. Because of $p(a) \geq p(c)$ and $y(\delta_D^-(a)) = 0$, we can assume by a flow argument similar to that of Lemma 12 that $y((r,c)) = 0$ holds. Similarly, we can assume that there is a flow f_b^c from r to c with $v(f_b^c) = f_b^c((r,b)) = y((r,b))$ and $f_b^c \leq y$—otherwise, we decrease $y((r,b))$ and increase $y((r,a))$. Let f_a^b and f_a^c be maximum flows from a to b and c with $f_a^b \leq y$ and $f_a^c \leq y$. If $v(f_a^b) = v(f_a^c) = 0$, we are effectively in the case $|T_p| \leq 2$, since we can restrict the problem to the support graph of (x, y).

So assume $v(f_a^b) > 0$ or $v(f_a^c) > 0$. Thus, $x(a) > 0$. Suppose $x(b) < 1$ and $x(c) < 1$. Note that either $v(f_a^b) = 0$, or both $v(f_a^b) > 0$ and $v(f_a^c) > 0$. First, suppose $v(f_a^b) = 0$. Thus, $y(\delta^-(b)) = 0$ and

$$\sum_{v \in V \setminus \{a,b\}} p(v) y(\delta_D^-(v)) > 0. \tag{68}$$

Define $\kappa := \frac{x(a)}{v(f_a^c)}$. Define $\tilde{y} \in \mathbb{R}^{A_r^+}$ by

$$\tilde{y}(e) := \max\{y(e), \kappa f_a^c(e)\}, \tag{69}$$

for all $e \in A_r^+$, and define $\tilde{x} \in \mathbb{R}^V$ accordingly. We have $\kappa x(c) = \tilde{x}(c)$. Thus, (68) implies $p^T \tilde{x} > p^T x$.

Second, suppose $v(f_a^b) > 0$ and $v(f_a^c) > 0$. In this case, (67) holds. Furthermore, $y(\delta_D^-(b)) = f_a^b(\delta_D^-(b))$. Thus, we can proceed as before and multiply both f_a^b and f_a^c by some $\kappa > 1$ to get a better LP solution. Overall, we have shown that $x(b) = 1$ or $x(c) = 1$. □

(Proof of Lemma 15 continued.) Let $\{a, b, t\} := T_p$ such that $x(t) = 1$ (which we can assume by Claim 1). Assume $y((r,t)) = x(t)$. In the following we mostly ignore the arcs $\delta^+(r)$ and concentrate on the bidirected graph $D = (V, A)$. We need to show that $x(a), x(b) \in \{0, 1\}$.

Suppose $x(a) \in (0,1)$ or $x(b) \in (0,1)$. Let f^a and f^b be maximum flows from t to a and b with capacity $y(e)$ for each arc $e \in A$. Note that $x(a) = 1$ or $x(b) = 1$; otherwise we could increase f^a, f^b, and y as in the proof of Claim 1. Assume w.l.o.g. $x(a) = 1$. Let $\overline{k} \in \mathbb{R}^A$ with $\overline{k}(e) := \max\{0, f^a(e) - f^b(e)\}$ for all $e \in A$. Let $\check{f}^a$ be a maximum t-a flow with $\check{f}^a \leq \overline{k}$.

Claim 2. *It holds that $v(\check{f}^a) > 0$.*

Proof. Suppose $v(\check{f}^a) = 0$. Let $\overline{V}_t \subset V$ be the set of vertices that can be reached by a directed path from t on the support graph induced by $\overline{k}$ (i.e., via arcs $e \in A$ with $\overline{k}(e) > 0$). Because of $v(\check{f}^a) = 0$, we have $a \notin \overline{V}_t$. Because (x, y) is optimal, we have $b \notin \overline{V}_t$. Because of $a \notin \overline{V}_t$ and $x(a) = 1$ we obtain

$$f^a(\delta^+(\overline{V}_t)) - f^a(\delta^-(\overline{V}_t)) = 1. \tag{70}$$

From the definition of $\overline{V}_t$ we get

$$f^b(\delta^+(\overline{V}_t)) \geq f^a(\delta^+(\overline{V}_t)). \tag{71}$$

Finally, because of $x(b) < 1$ we have

$$f^b(\delta^+(\overline{V}_t)) - f^b(\delta^-(\overline{V}_t)) < 1. \tag{72}$$

From (70),(71), and (72) we get

$$f^b(\delta^-(\overline{V}_t)) > f^a(\delta^-(\overline{V}_t)). \tag{73}$$

Thus, there is a $u \in \overline{V}_t \setminus \{t\}$ and $e_0 \in \delta^-(u)$ with $f^b(e_0) > f^a(e_0)$; note that $y(e_0) = f^b(e_0)$. By definition of $\overline{V}_t$, there is a directed path P from t to u such that $f^a(e) > f^b(e)$ for all $e \in A(P)$. Let $U \subset V$ with $e_0 \in \delta^-(U)$. If $b \in U$, the existence of P implies $y(\delta^-(U)) > x(b)$ (since we can increase f^b along P). If $a \in U$, we obtain from $y(e_0) > f^a(e_0)$ that $y(\delta^-(U)) > x(a)$. Thus, we can decrease $y(e_0)$ while staying feasible—in contradiction to the minimality or optimality of (x, y). □

(Proof of Lemma 15 continued.) We assume $v(\check{f}^a) < 1$. Otherwise the proof would already be complete, since the support graphs of f^a and f^b would be arc disjoint. Let $\hat{f}^a := f^a - \check{f}^a$. Further, for all $e \in A$ define $\tilde{f}^b(e) := \max\{0, f^b(e) - f^a(e)\}$. Note that $\hat{f}^a$ is a flow, but $\tilde{f}^b$ in general not. We further have

$$\sum_{v\in V} p(v)(\hat{f}^a + \check{f}^a + \tilde{f}^b)(\delta_D^-(v)) = \sum_{v\in V\setminus\{t\}} p(v)x(v). \tag{74}$$

Claim 3. *It holds that*

$$\frac{1}{v(\hat{f}^a)} \sum_{v\in V} p(v)(\hat{f}^a + \tilde{f}^b)(\delta_D^-(v)) = \frac{1}{v(\check{f}^a)} \sum_{v\in V} p(v)\check{f}^a(\delta_D^-(v)). \tag{75}$$

Proof. Because (x, y) is optimal and $f^a = \hat{f}^a + \check{f}^a$, we obtain from (74) that for any sufficiently small $\epsilon > 0$:

$$\big(1 + \frac{\epsilon}{v(\hat{f}^a)}\big) \sum_{v\in V} p(v)(\hat{f}^a + \tilde{f}^b)(\delta_D^-(v)) + \big(1 - \frac{\epsilon}{v(\check{f}^a)}\big) \sum_{v\in V} p(v)\check{f}^a(\delta_D^-(v)) \le \sum_{v\in V\setminus\{t\}} p(v)x(v), \tag{76}$$

and

$$\big(1 - \frac{\epsilon}{v(\hat{f}^a)}\big) \sum_{v\in V} p(v)(\hat{f}^a + \tilde{f}^b)(\delta_D^-(v)) + \big(1 + \frac{\epsilon}{v(\check{f}^a)}\big) \sum_{v\in V} p(v)\check{f}^a(\delta_D^-(v)) \le \sum_{v\in V\setminus\{t\}} p(v)x(v). \tag{77}$$

Note that on the right hand sides of the previous two inequalities we have essentially shifted an ϵ of the flow f^a from $\check{f}^a$ to $\hat{f}^a$, and vice-versa. Finally,

$$0 \overset{(76)}{\le} \frac{\epsilon}{v(\hat{f}^a)} \sum_{v\in V} p(v)(\hat{f}^a + \tilde{f}^b)(\delta_D^-(v)) - \frac{\epsilon}{v(\check{f}^a)} \sum_{v\in V} p(v)\check{f}^a(\delta_D^-(v)) \overset{(75)}{\le} 0, \tag{78}$$

which proves (75). □

(Proof of Lemma 15 continued.) Set

$$\kappa := \big(1 + \frac{v(\check{f}^a)}{v(\hat{f}^a)}\big). \tag{79}$$

Because of (75), and (74) we obtain

$$\kappa \sum_{v\in V} p(v)(\hat{f}^a + \tilde{f}^b)(\delta_D^-(v)) = \sum_{v\in V\setminus\{t\}} p(v)x(v). \tag{80}$$

Define $\tilde{y} \in \mathbb{R}^{A_r^+}$ by

$$\tilde{y}(e) := \max\{y(e), \kappa\hat{f}^a(e), \kappa f^b(e)\}, \tag{81}$$

for all $e \in A_r^+$, and define $\tilde{x} \in \mathbb{R}^V$ accordingly. It holds that $p^T\tilde{x} = p^T x$, and $\tilde{x}(c) = 1$. □

As the last piece, we have the now familiar contraction result (with a slight generalization).

Proposition 16. *$v_{LP}(ESA)$ is invariant under the contraction of adjacent vertices of non-negative weight.*

The proposition can be proven in a similar way as Proposition 5, with a few additional technical details. We now reach the main result of this section.

Theorem 17. *If $\alpha(G) \leq 3$, then $proj_x(\mathcal{P}_{LP}(ESA))$ is integral.*

Proof. Let $I = (G, p)$ be an MWCSP with $\alpha(G) \leq 3$. Let $I' = (V', E', p')$ be the MWCSP obtained from I by contracting all adjacent vertices of non-negative weight. Let A' be the bidirected equivalent of E'. Proposition 16 implies that $v_{LP}(ESA(I)) = v_{LP}(ESA(I'))$. Also, I' satisfies $|T'_p| \leq 3$ and the vertices T'_p are independent. By Lemma 15 and Lemma 12 there is an optimal LP solution $(\tilde{x}, \tilde{y})$ to $ESA(I')$ such that $x(u) \in \{0, 1\}$ for all $u \in T'_p$, and $y((r, t)) = 1$ for one $t \in T'_p$. Consider the RMWCSP $I'_t = (V', E', T'_f, p')$ with $T'_f := \{t\}$. For simplicity, we deviate from the assumption that fixed terminals have 0 weight. It holds that $v(ESA(I')) = v(RSA(I'_t))$ and $v_{LP}(ESA(I')) = v_{LP}(RSA(I'_t))$. We will show that

$$v_{LP}(RSA(I'_t)) = v(RSA(I'_t)), \tag{82}$$

which concludes the proof. Let (x, y) be the restriction of $(\tilde{x}, \tilde{y})$ to (V', A'). Note that (x, y) is an optimal LP solution to $RSA(I'_t)$. Suppose that (82) does not hold. Thus, by Lemma 3 there is a $v \in V' \setminus (T'_p \cup T'_f)$ with

$$y(\delta^+(v)) < y(\delta^-(v)). \tag{83}$$

The case $|T'_p| < 3$ can be readily ruled out by a flow argument. So assume $|T'_p| = 3$. Because of $\alpha(G) \leq 3$, at least one vertex $u \in T'_p$ is adjacent to v. Recall that $x(u) \in \{0, 1\}$. If $x(u) = 0$, we reduce the problem to the support graph of (x, y), which corresponds to the case $|T'_p| < 3$. So assume $x(u) = 1$. If $u \neq t$, define the RMWCSP $I'_u = (V', E', T''_f, p')$ with $T''_f := \{t, u\}$. Further, construct an optimal solution $(x, \tilde{y})$ to I'_u with root u analogously to Lemma 12. In this way, $\tilde{y}(\delta^+(v)) < \tilde{y}(\delta^-(v))$ holds again (for the same v as above). In the following, assume $u = t$. Define a new LP solution (x', y') from y as follows. For $a_0 := (t, v)$ set $y'(a_0) := y(\delta^+(v))$. For any $a \in \delta^-(v) \setminus \{a_0\}$ set $y'(a) := 0$. For all $a \in A' \setminus \delta^-(v)$ set $y'(a) := y(a)$. Set $x'(v) := y(\delta^+(v))$, and $x'(w) := x(w)$ for all $w \in V \setminus \{v\}$. By construction of I'_t it holds that $p(v) < 0$ (otherwise, v would have been contracted into u). Thus, $p'^T x' > p'^T x$. The feasibility of (x', y') can be seen as in the proof of Theorem 6. □

Note that there are graphs with $a(G) = 4$, such that $proj_x(\mathcal{P}_{LP}(ESA))$ is not integral. For an example, extend the graph in Figure 2 as follows. Add a new vertex v and edges between v and the (three) vertices of negative weight shown in the figure.

3.3 Comparison of the formulations

A result from Álvarez-Miranda et al. [2013a] states that the directed equivalents of ESA and $NCut$ induce the same polyhedral relaxation of the directed connected subraph polytope. This result suggest that the same relation holds for the undirected case. Unfortunately, the result from Álvarez-Miranda et al. [2013a] is not correct (the proof suffers from a similar problem as that discussed in Appendix A for the rooted case). The strict inclusion result given in the next proposition can indeed also be extended to the directed case.

Proposition 18. *The following inclusion holds and can be strict:*

$$proj_x(\mathcal{P}_{LP}(ESA)) \subset \mathcal{P}_{LP}(NCut). \tag{84}$$

Proof. Let $(x, y) \in \mathcal{P}_{LP}(ESA)$ and let $a, b \in V$, $a \neq b$. Let $C \in \mathcal{C}(a, b)$ and let U_a be the connected component in the graph $(V \setminus C, E[V \setminus C])$ with $a \in U_a$. Define $\bar{U}_b := V \setminus U_a$ and $\bar{U}_a := U_a \cup C$. Because of $\bar{U}_a \cap \bar{U}_b = C$, one obtains

$$y(\delta^-(\bar{U}_a)) + y(\delta^-(\bar{U}_b)) = y(\delta^-(\bar{U}_a \cup \bar{U}_b)) + y(\delta^-(C)), \tag{85}$$

where we use $\delta^- := \delta^-_{D_r}$. Thus,

$$x(a) + x(b) \overset{(60)}{\leq} y(\delta^-(\bar{U}_a)) + y(\delta^-(\bar{U}_b)) \tag{86}$$

$$\overset{(85)}{=} y(\delta^+(r)) + y(\delta^-(C)) \tag{87}$$

$$\overset{(61)}{\leq} 1 + x(C). \tag{88}$$

An example for a strict inclusion is given in Figure 2. E.g., consider the following point that is in $\mathcal{P}_{LP}(NCut)$, but not in $proj_x(\mathcal{P}_{LP}(ESA))$: Set the values of all negative weight node variables to 0.5 and the remainder to 1. □

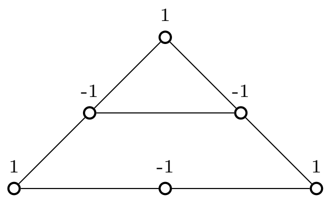

Figure 2: MWCSP instance with given node weights.

Next, we consider the indegree constraints. Following Wang et al. [2017], we define

$$\mathcal{Q}' := \{x \in \mathbb{R}^V_{\geq 0} \mid \text{x satisfies all indegree constraints}\}. \tag{89}$$

While $\mathcal{Q}' \nsubseteq \mathcal{P}_{LP}(NCut)$, see e.g. Wang et al. [2017], the indegree constraints cannot improve the ESA formulation, as the following proposition shows.

Proposition 19. *The following inclusion holds and can be strict:*

$$proj_x(\mathcal{P}_{LP}(ESA)) \subset \mathcal{Q}'. \tag{90}$$

Proof. Consider an undirected graph G, and let D be its bidirected equivalent. Furthermore, let D_r be the extended, directed graph on which ESA is defined. Let $(x, y) \in \mathcal{P}_{LP}(ESA)$. First, note that constraints (59) and (60) imply for all $\{v, w\} \in E$ that

$$\min\{y(\delta^-_{D_r}(v)), y(\delta^-_{D_r}(w))\} \geq y((v, w)) + y((w, v)). \tag{91}$$

Let d be an indegree vector. It holds that

$$\begin{aligned}
\sum_{v \in V} x(v) &= \sum_{a \in A_r} y(a) \\
&\leq \sum_{a \in A} y(a) + 1 \\
&= \sum_{\{v,w\} \in E} \big(y((v,w)) + y((w,v)) \big) + 1 \\
&\overset{(91)}{\leq} \sum_{\{v,w\} \in E} \min\{ y(\delta^-_{D_r}(v)), y(\delta^-_{D_r}(w)) \} + 1 \\
&\leq \sum_{v \in V} d_v x(v) + 1,
\end{aligned}$$

which implies that (57) is satisfied by x; thus, $x \in \mathcal{Q}'$. For a strict inclusion consider the graph in Figure 2 and the point x as defined in the proof of Proposition 18. □

Summarizing the results of this section, one obtains:

Theorem 20. *It holds that*

$$proj_x(\mathcal{P}_{LP}(ESA)) \subset \mathcal{Q}' \cap \mathcal{P}_{LP}(NCut), \tag{92}$$

and the inclusion can be strict.

Finally, note that by using one flow for each vertex, similar to the DF formulation, it is also possible to obtain a compact extended formulation for the connected subgraph polytope that is equivalent to ESA—and thus (strictly) stronger than the combined node-separator and indegree formulation.

4 Algorithms

The previous two sections have established theoretical results that suggest the use of edge based formulations as opposed to node based ones for the solution of MWCSP and related problems. A natural candidate for the practical solution of IP formulations with exponentially many constraints is branch-and-cut. However, the practical success of a branch-and-cut algorithm also highly depends on additional methods such as presolving techniques or (primal and dual) heuristics. Thus, this section introduces several algorithms for MWCSP, that can be used as part of a branch-and-cut algorithm based on ESA^+_{FB} as described in Rehfeldt and Koch [2019].

We will only consider the non-rooted MWCSP in this section for the sake of simplicity. All MWCSP algorithms can be adapted for RMWCSP by setting sufficiently high weights for each fixed terminal.

4.1 A primal heuristic

A common greedy heuristic for SPG, see Takahashi and A. [1980], and also already used for MWCSP, see Rehfeldt and Koch [2019], is to start with a single vertex and iteratively connect a new terminal, or (in the case of MWCSP) a vertex of positive weight by a path to the current tree. In the case of the SPG a natural choice for the connection is a shortest path, but in the case of MWCSP the choice is less clear. The following algorithm chooses paths that also take intermediary vertices of positive weight into account.

Let vertex $v_r \in V$ be the start vertex. Initially, define for all $v \in V \setminus \{v_r\}$: $p^+(v) := \max\{p(v), 0\}$, $p^-(v) := \max\{-p(v), 0\}$, $\tilde{d}(v) := \infty$. For v_r set all these values to 0. Define a predecessor $pred(v) := null$ for each $v \in V$. Define the initial tree S as $V(S) := \{v_r\}, E(S) := \emptyset$, and set $Q := \{v\}$. While $Q \neq \emptyset$ proceed as follows. Choose a $v = \arg\min_{u \in Q} \tilde{d}(u)$ and remove v from Q. If $p(v) > \tilde{d}(v)$, add the path P between S and v marked by $pred$ to S. Further, set for each $w \in V(P)$: $\tilde{d}(w) := p^+(w) := 0$, and add w to Q.

In any case, define for each $\{v, w\} \in \delta(v)$

$$\tilde{d}_{vw} := \tilde{d}(v) + p^-(w) - \min\{p^-(w), p^+(v)\}. \tag{93}$$

If $\tilde{d}_{vw} < \tilde{d}(w)$, add w to Q and set $\tilde{d}(w) := \tilde{d}_{vw}$, $pred(w) := v$. Note that in (93) we cannot just subtract $p^+(v)$, because otherwise the algorithm might cycle.

Once Q is empty, we compute a spanning tree on the graph induced by $V(S)$, while trying to have vertices $V(S) \setminus T_p$ as leaves. Then, we use a linear-time dynamic programming algorithm described in Magnanti and Wolsey [1995] to compute a maximum-weight connected subgraph on this tree.

Finally, note that the above heuristic can also be extended to the (more general) prize-collecting Steiner tree problem.

4.2 A dual heuristic

In Wong [1984] a dual-ascent algorithm for *DCut* was introduced, and has since then be used as important component for reduction techniques, see e.g. Duin [1993], or initial cut generation, see e.g. Polzin and Daneshmand [2001b]. Consider an SAP (V, A, T, r, c). Let $\mathcal{W} := \{W \subset V \mid r \notin W, T \cap W \neq \emptyset\}$. Consider the dual of *DCut*:

$$\max \sum_{W \in \mathcal{W}} \mu_W \tag{94}$$

$$\text{s.t.} \sum_{W \in \mathcal{W} | a \in \delta^-(W)} \mu_W \leq c(a) \qquad \text{for all } a \in A, \tag{95}$$

$$\mu \geq 0. \tag{96}$$

Given a dual solution μ, let $A_\mu \subseteq A$ be the set of arcs for which (95) is tight. For each $t \in T \setminus \{r\}$, define the *root component* U_t of t as the set of vertices $v \in V$ such that there exists a directed v-t path in A_μ. A root component U_t is *active* if $T \cap U_t = \{t\}$. Initially, the dual-ascent algorithm sets $\mu := 0$. In each iteration an active root component U_t is chosen and μ is increased until (95) becomes tight for at least one $a \in \delta^-(U_t)$. This increase can be done implicitly by just adapting the reduced costs. The algorithm terminates when no active root component is left.

In Rehfeldt et al. [2019] dual-ascent was used for MWCSP based on the following transformation of MWCSP to SAP:

Transformation 1 (MWCSP to SAP)**.**
Input: *An MWCSP $I = (V, E, p)$*
Output: *An SAP $I' = (V', A', T', c', r')$*

1. *Set $V' := V$, $A' := \{(v, w) \in V' \times V' \mid \{v, w\} \in E\}$, $s := |T_p|$.*
2. *Set $c' : A' \to \mathbb{R}_{\geq 0}$ such that for $a = (v, w) \in A'$:*
$$c'(a) = \begin{cases} -p(w), & \text{if } p(w) < 0 \\ 0, & \text{otherwise} \end{cases}$$

3. *Add two vertices* r' *and* v'_0 *to* V'.

4. *Denote the set of all* $v \in V$ *with* $p(v) > 0$ *by* $T = \{t_1, ..., t_s\}$ *and define* $M := \sum_{t \in T_p} p(t)$.

5. *For each* $i \in \{1, ..., s\}$:

   (a) *Add an arc* (r', t_i) *of weight* M *to* A'.

   (b) *Add a new node* t'_i *to* V'.

   (c) *Add arcs* (t_i, v'_0) *and* (t_i, t'_i) *to* A', *both being of weight* 0.

   (d) *Add an arc* (v'_0, t'_i) *of weight* $p(t_i)$ *to* A'.

6. *Define the set of terminals* $T' := \{t'_1, ..., t'_s\} \cup \{r'\}$.

7. ***Return*** (V', A', T', c', r').

Similar to the previous section, one shows that applying the *DCut* formulation on the SAP I' from the above transformation yields (after a constant shift of the objective) the same optimal LP value as ESA. Furthermore, one notes that the constraints (2) for I' corresponding to all non-zero μ_W can be readily transformed to constraints (60) for ESA^+. These can be used as initial constraints for a branch-and-cut algorithm.

In practice, one tries to only increase small root components in the dual-ascent algorithm. Moreover, it is advantageous to only update the currently used root component and rebuild U_t by a BFS or DFS after each change of t, see Pajor et al. [2017]. For I' one notices that for distinct terminals t_i, t_j with $v'_0 \in U_{t_i}$ and $v'_0 \in U_{t_j}$ it holds that $U_{t_i} \setminus \{t_i\} = U_{t_j} \setminus \{t_j\}$. However, due to the structure of I' all root components will remain active until the end of the algorithm. Thus, a simple, but sometimes highly effective modification of I' is to make v'_0 a terminal. In this way, any root component U_t ceases to be active as soon as $v'_0 \in U_t$. Still, the final reduced costs remain the same.

4.3 Reduction techniques

Reduction techniques are a vital component for practical exact solution of MWCSP, see Leitner et al. [2018], Rehfeldt and Koch [2019], and also for many other combinatorial optimization problems, such as SPG, see Polzin and Daneshmand [2001b] or maximum clique, see Verma et al. [2015]. Reduction techniques for MWCSP (and other combinatorial optimizations problems) aim to transform a given problem instance I to an instance I' such that any optimal solution S' to I' can be transformed to an optimal solution S to I, and I' is (hopefully) easier to solve than I. Typically, reduction techniques aim to delete edges or vertices; we already saw the example of positive weight neighbors contraction in the previous section.

Note that an optimal solution may consist of a single vertex. Thus, care needs to be taken to avoid the deletion or modification of an optimal positive weight vertex. Indeed, one finds wrong reduction tests in literature as detailed in Rehfeldt et al. [2019]. An example is the contraction of a positive weight vertex of degree 1 into its (negative weight) neighbor. Also, just remembering the maximal-weight connected vertex before the start of the reduction techniques, as suggested in the literature, is not sufficient: A single vertex solution might be created during the reduction process on a reduced problem. Indeed, guarding measures during the reduction process cannot be avoided due to the following proposition. It can be readily proven by a reduction from the decision variant of MWCSP.

Proposition 21. *Deciding whether a single-vertex maximum-weight connected subgraph exists is* $\mathcal{NP}$*-complete.*

The remainder of this section introduces two new reduction methods for MWCSP—these techniques can also be applied to RMWCSP if sufficiently high weights for each fixed terminal are used.

Bottleneck distances

Bottleneck Steiner distances are a classic concept for SPG reduction techniques introduced by Duin and Volgenant [1989]. Initially, this section translates (in a straightforward way) a recent generalization of this concept for PCSTP by Rehfeldt and Koch [2020] to MWCSP. Let $v, w \in V$. A finite walk $W = (v_1, e_1, v_2, e_2, ..., e_r, v_r)$ with $v_1 = v$ and $v_r = w$ will be called *positive-weight constrained* (v, w)*-walk* if no $u \in T_p \cup \{v, w\}$ is contained more than once in W. For any $k, l \in \mathbb{N}$ with $1 \leq k \leq l \leq r$ define the subwalk $W(v_k, v_l) := (v_k, e_k, v_{k+1}, e_{k+1}, ..., e_l, v_l)$. In the following, let W be a positive-weight constrained (v, w)-walk. Define the *interior cost* of W as:

$$C^-(W) := \sum_{u \in V(W) \setminus \{v,w\}} p(u), \tag{97}$$

where the convention that the empty sum equals 0 is assumed, so the interior cost of an edge is likewise 0. Furthermore, define the *positive-weight constrained length* of W as:

$$l_{pw}(W) := \min\{C^-(W(v_k, v_l)) \mid 1 \leq k \leq l \leq r,\ v_k, v_l \in T_p \cup \{v, w\}\}. \tag{98}$$

Note that $l_{pw}(W) \leq 0$ holds, because the interior cost of an edge is 0. Denote the set of all positive-weight constrained (v, w)-walks by $\mathcal{W}_{pw}(v, w)$ and define the *positive-weight constrained distance* between v and w as

$$d_{pw}(v, w) := \max\{l_{pw}(W) \mid W \in \mathcal{W}_{pw}(v, w)\}. \tag{99}$$

For any $U \subseteq V$ denote by K_U the complete graph on U. Furthermore, define for each edge $\{v, w\}$ of K_U weights $d^U_{pw}(v, w) := d_{pw}(v, w)$—with d_{pw} being the positive-weight constrained distance in the original MWCSP. Based on these concepts, we introduce two new edge elimination criteria in the following. Both of them dominate the NPV$_k$ test from Rehfeldt et al. [2019]—also if the (weaker) bottleneck distance from Rehfeldt et al. [2019] is used.

Proposition 22. *Let $e = \{v, w\} \in E$ with $p(v) \leq 0$. Define $\Delta := (T_p \cup \{w\}) \cap N(v)$, and*

$$\mathcal{U} = \big\{U \subseteq N(v) \mid |U| \geq 2, U \supseteq \Delta\big\}.$$

If for all $U \in \mathcal{U}$ the weight of a minimum spanning tree on $(K_U, -d^U_{pw})$ is smaller than $-p(v)$, then at least one optimal solution does not contain edge e.

Proof. Let S be a connected subgraph with $e \in E(S)$. We will show that there is a connected subgraph S' with $e \notin E(S')$ such that

$$P(S) \leq P(S'). \tag{100}$$

We can assume that S is a tree, and that $\Delta \subseteq N_S(v)$. Otherwise, we can modify S to satisfy these conditions and still contain e without decreasing the weight of S. Note that if v is of degree 1 in S, we can simply delete edge e to obtain the desired S'. So assume $|N_S(v)| \geq 2$.

Let $U := N_S(v)$. Let $\hat{S}$ be the subgraph obtained from S by removing vertex v and all incident edges. Let $S^{(1)}, ..., S^{(k)}$ be the (inclusion-wise maximal) connected components of $\hat{S}$. Note that $k = |U|$. Let F_U be a minimum spanning tree on $(K_U, -d^U_{pw})$ and denote its weight by

$-C_U$ (recall that d_{pw} is non-positive). I.e., C_U is a maximum-weight spanning tree on (K_U, d^U_{pw}). By the assumption of the proposition it holds that

$$C_U - p(v) > 0. \tag{101}$$

Assume that the $S^{(i)}$ are ordered such that for each $i \in \{2, ..., k\}$ there are vertices $q^{(i)} \in V(\bigcup_{h \leq i} S^{(h)}) \cap U$ and $r^{(i)} \in V(S^{(i+1)}) \cap U$ such that there is a $(q^{(i)}, r^{(i)})$-walk $W^{(i)}$ in (V, E) corresponding to an edge in the spanning tree F_U. Note that $l_{pw}(W^{(i)}) = d_{pw}(q^{(i)}, r^{(i)})$. Set $\hat{S}^{(1)} := S^{(1)}$ and proceed for $i = 2, ..., k$ as follows. First, observe that $v \notin V(W^{(i)})$ due to the assumptions of the proposition. Let $v_1, v_2, ..., v_s$ be the vertices encountered (in this order) when traversing $W^{(i)}$ from $q^{(i)}$ to $r^{(i)}$. So in particular $v_1 = q^{(i)}$ and $v_s = r^{(i)}$. Let b be the minimum number such that $v_b \in V(S^{(i)})$. Further, let a be the largest number in $\{1, 2, .., b\}$ such that $v_a \in V(\hat{S}^{(i-1)})$. Further, define $x := \max\{j \in \{1, ..., a\} \mid v_j \in T_p \cup \{v_1\}\}$ and $y := \min\{j \in \{b, ..., s\} \mid v_j \in T_p \cup \{v_s\}\}$. By definition, $x \leq a < b \leq y$ and furthermore:

$$C^-(W^{(i)}(v_a, v_b)) \geq C^-(W^{(i)}(v_x, v_y)) \geq l_{pw}(W^{(i)}) = d_{pw}(q^{(i)}, r^{(i)}). \tag{102}$$

Define $\hat{S}^{(i)} := \hat{S}^{(i-1)} \cup S^{(i)} \cup W^{(i)}(v_a, v_b)$, with a slight abuse of notation $W^{(i)}(v_a, v_b)$ is considered as a subgraph here. Ultimately, $S' := \hat{S}^{(k)}$ is a connected subgraph and it holds that

$$P(S') \overset{(102)}{\geq} P(\hat{S}) + C_U = P(S) + C_U - p(v) \overset{(101)}{>} P(S). \tag{103}$$

Because $v \notin V(S')$ implies $e \notin E(S')$, the proposition is proven. □

Another reduction test can be obtained by splitting the neighborhood of the edge considered for elimination, as detailed in the following proposition.

Proposition 23. *Let $e = \{v, w\} \in E$. Assume that $p(v) + p(w) \leq 0$. Define $\Delta := (T_p \cap N(e))$. Further, define*

$$\mathcal{U} = \big\{U \subseteq N(e) \mid U \supseteq \Delta, U \cap (N(v) \setminus \{w\}) \neq \emptyset, U \cap (N(w) \setminus \{v\}) \neq \emptyset\big\}.$$

If for all $U \in \mathcal{U}$ the weight of a minimum spanning tree on $(K_U, -d^U_{pw})$ is smaller than $-(p(v) + p(w))$, then no optimal solution contains edge e.

The proposition can be proved in a similar way to the previous one. Note that both proposition can be extended to the case of equality if the walks corresponding to positive-weight constrained distances do not contain edge e.

5 Computational experiments

The algorithms introduced in the previous section have been integrated into the branch-and-cut Steiner tree framework SCIP-Jack, see Gamrath et al. [2017], which includes a state-of-the-art MWCSP solver, see Rehfeldt and Koch [2019]. The IP formulation used by the solver is (essentially) ESA^+_{FB}. To evaluate the practical impact of the new algorithms, we will compare the new version of SCIP-Jack with the one described in Rehfeldt and Koch [2019]. We will refer to the latter as SCIP-Jack-base. Furthermore, this section also serves to show the practical strength of ESA^+ (and its rooted equivalents) on hard benchmark instances, where otherwise decisive techniques such as reduction methods have a smaller impact.

The computational experiments were performed on Intel Xeon CPUs E3-1245 with 3.40 GHz and 32 GB RAM. CPLEX 12.8 IBM was employed as underlying LP solver. All results were obtained single-threaded. Five benchmark test sets from the literature and the 11th DIMACS Challenge are used, as detailed in Table 1. The test set EASY is the union of the test sets ACTMOD, SHINY, and JMPALMK. The test sets PUCN, PUCNU, and HAND contain originally USPG and prize-collecting USPG instances, which have been transformed to RMWCSP (for USPG) and MWCSP.

Name	Instances	$\|V\|$	$\|E\|$	Status	Description
EASY	119	232-5226	202-93394	solved	Collection of easy real-world and random instances, see Dittrich et al. [2008], Loboda et al. [2016].
HANDS	20	39600 - 42500	78704 - 84475	solved	Images of hand-written text derived from a signal processing problem see Hegde et al. [2014].
HANDB	28	158400 - 169800	315808 - 338551	unsolved	
PUCN	13	64-4096	192-24574	unsolved	Hard USPG instances from the 11th DIMACS Challenge DIMACS.
PUCNU	18	64-4096	192-24574	unsolved	Hard prize-collecting USPG instances from the 11th DIMACS Challenge DIMACS.

Table 1: Classes of MWCSP instances.

Table 2 provides aggregated results of the experiments with a time limit of two hours. The first column shows the test set considered in the current row. Columns two and three show the shifted geometric mean from Achterberg [2007] (with shift 1) of the run time taken by the respective solver. Column four shows the speedup of the new solver, measured as the ratio of the previous two values. The next three columns provide the same information with respect to the maximum run time. The last two columns give the number of instances solved by the respective solver. The first test set, can be solved by both versions of SCIP-Jack in less than 0.3 seconds. On the next two test sets, the new solver is thrice and twice as fast with respect to the shifted geometric mean. On the last two test sets, the new solver is again significantly faster, and additionally solves three more instances to optimality.

Table 2: Computational comparison of SCIP-Jack-base, denoted by *base* and the solver described in this article, denoted by *new*.

		mean time			maximum time			# solved	
Test set	#	base [s]	new [s]	speedup	base [s]	new [s]	speedup	base	new
EASY	119	0.0	0.0	1.0	0.3	**0.2**	**1.5**	119	119
HANDS	20	0.6	**0.2**	**3.0**	3.2	**1.6**	**2.0**	20	20
HANDB	28	8.9	**4.4**	**2.0**	>7200	>7200	1.0	26	26
PUCN	13	72.1	**59.5**	**1.2**	>7200	>7200	1.0	8	**9**
PUCNU	18	124.5	**57.7**	**2.2**	>7200	>7200	1.0	12	**14**

Note that already the base version of SCIP-Jack is significantly faster than other MWCSP solvers described in the literature; see Rehfeldt and Koch [2019] for more details. Finally, Table 3 shows results obtained with a time limit of 24 hours on previously unsolved instances from the 11th DIMACS Challenge. We provide the final optimality gap (column two), the best primal objective value (column three), and the known primal objective value from the literature (column four). Overall, five instances can be solved for the first time to optimality. Apart from *handbd13*, optimality is reached for all these instances within the previous 2 h time limit. The instances *cc3-10n*, *cc3-11n*, and *cc3-12n* can be solved by both the new SCIP-Jack and SCIP-Jack-base. However, for the remaining instances SCIP-Jack-base cannot reach the new primal bounds.

The MWCSP solver developed for this article will be made publicly available as part of the next release of the SCIP-Jack software package.

Table 3: Improvements on unsolved DIMACS instances in a long run (24 h). Solutions values are given w.r.t. the original USPG and prize-collecting USPG instances.

Name	gap [%]	new UB	previous UB
handbd13	**opt**	**13.184261**	13.18549
cc10-2n	**opt**	**179**	180
cc3-10n	**opt**	**75**	75
cc3-11n	**opt**	**92**	92
cc3-12n	**opt**	**111**	111
cc11-2n	1.0	324	327
cc12-2n	0.9	613	617
cc7-3n	1.0	287	289
cc9-2n	2.1	98	99

6 Conclusion

The first part of this article has analyzed node and edge based formulations for combinatorial optimization problems based on induced connectivity. Furthermore, we have shown conditions for the LP relaxations to be tight. In particular, a (compact) complete description of the connected subgraph polytope for graphs with less than four independent vertices has been given. Overall, it has been demonstrated that the edge based formulations consistently provide stronger LP relaxations than their node based counterparts. For MWCSP, the considered edge cut formulation has been shown to be strictly stronger than the combination of the well-known node-separator and indegree formulations. Motivated by these results, the second part of the article has introduced algorithms to enhance the practical exact solution of MWCSP and related problems by means of edge based IP formulations. Computational experiments have demonstrated that the resulting solver outperforms the previous state of the art.

7 Acknowledgements

The work for this article has been conducted within the Research Campus Modal funded by the German Federal Ministry of Education and Research (fund number 05M14ZAM), and has also been supported by the the DFG Cluster of Excellence MATH+.

References

T. Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007.

E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The maximum weight connected subgraph problem. In *Facets of Combinatorial Optimization*, pages 245–270. Springer Berlin Heidelberg, 2013a.

E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. *The Rooted Maximum Node-Weight Connected Subgraph Problem*, pages 300–315. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013b. ISBN 978-3-642-38171-3. doi: 10.1007/978-3-642-38171-3_20. URL http://dx.doi.org/10.1007/978-3-642-38171-3_20.

M. Ball, W. Liu, and W. Pulleyblank. Two terminal Steiner tree polyhedra. In *Contributions to Operations Research and Economics The Twentieth Anniversary of CORE*, pages 251–284. MIT Press, 1989.

J. Banfi. *Multirobot exploration of communication-restricted environments*. PhD thesis, Politecnico di Milano, 2018.

M. D. Biha, H. L. M. Kerivin, and P. H. Ng. Polyhedral study of the connected subgraph problem. *Discrete Mathematics*, 338(1):80–92, 2015. doi: 10.1016/j.disc.2014.08.026. URL http://dx.doi.org/10.1016/j.disc.2014.08.026.

A. Buchanan, J. S. Sung, S. Butenko, and E. L. Pasiliao. An integer programming approach for fault-tolerant connected dominating sets. *INFORMS Journal on Computing*, 27(1):178–188, 2015.

A. Buchanan, Y. Wang, and S. Butenko. Algorithms for node-weighted steiner tree and maximum-weight connected subgraph. *Networks*, 72(2):238–248, 2018. doi: 10.1002/net.21825.

R. Carvajal, M. Constantino, M. Goycoolea, J. P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Operations Research*, 61(4):824–836, 2013. doi: 10.1287/opre.2013.1183.

C. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 1274–1281, 2012. doi: 10.1109/CVPR.2012.6247811. URL http://dx.doi.org/10.1109/CVPR.2012.6247811.

J. Conrad, C. P. Gomes, W.-J. van Hoeve, A. Sabharwal, and J. Suter. Connections in networks: Hardness of feasibility versus optimality. In P. Van Hentenryck and L. Wolsey, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 16–28, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-72397-4.

B. Dilkina and C. P. Gomes. Solving connected subgraph problems in wildlife conservation. In *Proceedings of the 7th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, CPAIOR'10, pages 102–116, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-13519-6, 978-3-642-13519-4. doi: 10.1007/978-3-642-13520-0_14. URL http://dx.doi.org/10.1007/978-3-642-13520-0_14.

DIMACS. 11th DIMACS Challenge. http://dimacs11.zib.de/. Accessed: January 10. 2020.

M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. In *ISMB*, pages 223–231, 2008.

C. Duin. *Steiner Problems in Graphs*. PhD thesis, University of Amsterdam, 1993.

C. W. Duin and A. Volgenant. Reduction tests for the Steiner problem in graphs. *Networks*, 19 (5):549–567, 1989. ISSN 1097-0037. doi: 10.1002/net.3230190506.

B. Filipecki and M. Van Vyve. Stronger path-based extended formulation for the steiner tree problem. *Networks*, 75(1):3–17, 2020. doi: 10.1002/net.21901. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21901.

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. *Mathematical Programming Computation*, 9(2):203–229, Jun 2017. ISSN 1867-2957. doi: 10.1007/s12532-016-0111-0.

A. Fügenschuh and M. Fügenschuh. Integer linear programming models for topology optimization in sheet metal design. *Mathematical Methods of Operations Research*, 68(2):313 – 331, 2008.

G. Gamrath, T. Koch, S. Maher, D. Rehfeldt, and Y. Shinano. SCIP-Jack - A solver for STP and variants with parallelization extensions. *Mathematical Programming Computation*, 9(2): 231 – 296, 2017. doi: 10.1007/s12532-016-0114-x.

R. S. Garfinkel and G. L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B–495, 1970.

M. X. Goemans and Y. Myung. A catalog of Steiner tree formulations. *Networks*, 23(1):19–28, 1993. doi: 10.1002/net.3230230104.

C. Hegde, P. Indyk, and L. Schmidt. A fast, adaptive variant of the goemans-williamson scheme for the prize-collecting steiner tree problem. In *Workshop of the 11th DIMACS Implementation Challenge*. Workshop of the 11th DIMACS Implementation Challenge, 2014.

F. Hwang, D. Richards, and P. Winter. *The Steiner Tree Problem*. Annals of Discrete Mathematics. Elsevier Science, 1992.

IBM. Cplex. URL `https://www.ibm.com/analytics/cplex-optimizer`.

D. S. Johnson. The np-completeness column: An ongoing guide. *J. Algorithms*, 6(1):145–159, 1985. URL `http://dx.doi.org/10.1016/0196-6774(85)90025-2`.

T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32:207–232, 1998. doi: 10.1002/(SICI)1097-0037(199810)32:3\%3C207::AID-NET5\%3E3.0.CO;2-O.

T. Koch, A. Martin, and S. Voß. SteinLib: An updated library on Steiner tree problems in graphs. In D.-Z. Du and X. Cheng, editors, *Steiner Trees in Industries*, pages 285–325. Kluwer, 2001. ISBN 1-402-00099-5.

B. Korte, L. Lovász, and R. Schrader. *Greedoids*, volume 4. Springer Science & Business Media, 2012.

M. Leitner, I. Ljubic, M. Luipersbeck, and M. Sinnl. A dual ascent-based branch-and-bound framework for the prize-collecting steiner tree and related problems. *INFORMS Journal on Computing*, 30(2):402–420, 2018.

W. Liu. A lower bound for the steiner tree problem in directed graphs. *Networks*, 20(6):765–778, 1990. doi: 10.1002/net.3230200606. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230200606`.

A. A. Loboda, M. N. Artyomov, and A. A. Sergushichev. *Solving Generalized Maximum-Weight Connected Subgraph Problem for Network Enrichment Analysis*, pages 210–221. Springer International Publishing, Cham, 2016. ISBN 978-3-319-43681-4. doi: 10.1007/978-3-319-43681-4_17. URL `http://dx.doi.org/10.1007/978-3-319-43681-4_17`.

T. L. Magnanti and L. A. Wolsey. *Handbooks in Operations Research and Management Science*, volume Volume 7, chapter Chapter 9 Optimal trees, pages 503–615. Elsevier, 1995.

J. Moody and D. R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American sociological review*, pages 103–127, 2003.

J. Nederlof. Fast polynomial-space algorithms using inclusion-exclusion. *Algorithmica*, 65 (4):868–884, 2013. doi: 10.1007/s00453-012-9630-x. URL `https://doi.org/10.1007/s00453-012-9630-x`.

T. Pajor, E. Uchoa, and R. F. Werneck. A robust and scalable algorithm for the steiner problem in graphs. *Mathematical Programming Computation*, Sep 2017. ISSN 1867-2957. doi: 10.1007/s12532-017-0123-4. URL `https://doi.org/10.1007/s12532-017-0123-4`.

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

T. Polzin and S. V. Daneshmand. Improved algorithms for the steiner problem in networks. *Discrete Appl. Math.*, 112(1-3):263–300, Sept. 2001b. ISSN 0166-218X.

D. Rehfeldt and T. Koch. Combining NP-Hard Reduction Techniques and Strong Heuristics in an Exact Algorithm for the Maximum-Weight Connected Subgraph Problem. *SIAM Journal on Optimization*, 29(1):369–398, 2019. doi: 10.1137/17M1145963.

D. Rehfeldt and T. Koch. On the exact solution of prize-collecting steiner tree problems. Technical Report 20-11, ZIB, Takustr. 7, 14195 Berlin, 2020.

D. Rehfeldt, T. Koch, and S. J. Maher. Reduction techniques for the prize collecting Steiner tree problem and the maximum-weight connected subgraph problem. *Networks*, 73(2):206–233, 2019. doi: 10.1002/net.21857.

H. Takahashi and M. A. An approximate solution for the Steiner problem in graphs. *Mathematica Japonicae*, 24:573 – 577, 1980.

E. Uchoa. Reduction Tests for the Prize-collecting Steiner Problem. *Oper. Res. Lett.*, 34(4): 437–444, July 2006.

A. Verma, A. Buchanan, and S. Butenko. Solving the maximum clique and vertex coloring problems on very large sparse networks. *INFORMS Journal on Computing*, 27(1):164–177, 2015. doi: 10.1287/ijoc.2014.0618.

Y. Wang, A. Buchanan, and S. Butenko. On imposing connectivity constraints in integer programs. *Mathematical Programming*, pages 1–31, 2017. ISSN 1436-4646. doi: 10.1007/s10107-017-1117-8. URL `http://dx.doi.org/10.1007/s10107-017-1117-8`.

R. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28:271287, 1984.

A Node separators and rejoining of flows

Consider the directed RMWCSP instance (G, T, p, r) with $G = (V, A)$ depicted in Figure 3.

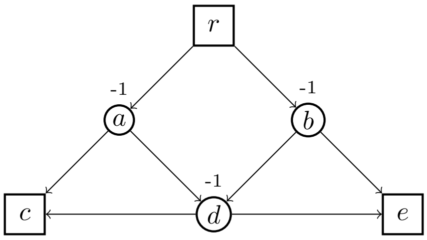

Figure 3: Directed RMWCSP instance.

A proof from Álvarez-Miranda et al. [2013b] intends to show that $v_{LP}(RNCut) \leq v_{LP}(\text{RSA})$ holds. For this purpose, the authors consider an arbitrary solution $\bar{x} \in \mathcal{P}_{LP}(\text{RNCut})$ and construct an auxiliary graph G' by replacing each node $v \in V \setminus \{r\}$ with an arc (v_1, v_2). All ingoing arcs of v become ingoing arcs of v_1, and all outgoing arcs of v are now outgoing arcs of v_2. Moreover, (non-negative) capacities k' on G' are introduced for each arc (v', w') of G' by

$$k'(v', w') := \begin{cases} \bar{x}(v), & \text{if } (v', w') = (v_1, v_2) \text{ for a } v \in V \\ 1, & \text{otherwise.} \end{cases}$$

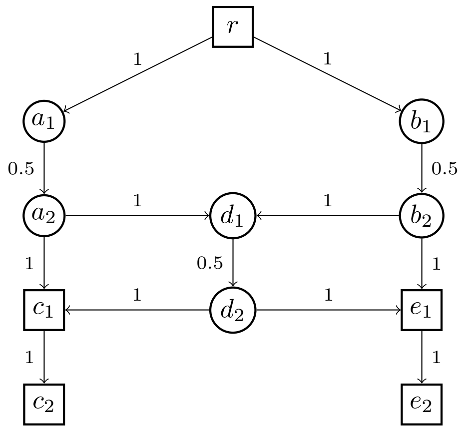

Figure 4: Illustration of an auxiliary support graph G' corresponding to the instance in Figure 3 regarding the optimal solution $\bar{x}(v) = 0.5$, $v \in V \setminus T$, and $\bar{x}(t) = 1$, $t \in T$, to the RNCut formulation.

Figure 4 shows an auxiliary support graph of the instance illustrated by Figure 3. It is possible to send a flow with flow value $\bar{x}(v)$ from root node r to each arc (v_1, v_2) with $v \in V \setminus \{r\}$ because of constraints (32). Let $f^v(j, l)$ be the amount of a flow with source node r, sink node $v \in V \setminus \{r\}$, and flow value $\bar{x}(v)$ sent along arc (j, l). Define the arc variables $\hat{y}(j, l)$, $(j, l) \in A$, of the RSA formulation as follows:

$$\hat{y}(j,l) := \begin{cases} \max_{v \in V \setminus \{r\}} f^v(j_2, l_1), & j, l \in V \setminus \{r\} \\ \max_{v \in V \setminus \{r\}} f^v(j, l_1), & j = r, l \in V \setminus \{r\}. \end{cases}$$

Hence, the arc variables of the instance in Figure 4 are given by $\hat{y}(j, l) = 0.5$ for each $(j, l) \in A$. Moreover, define the node variables as $\hat{x}(v) = \hat{y}(\delta^-(v))$. Thus, in our case, it holds $\hat{x}(a)$, $\hat{x}(b)$, $\hat{x}(c)$, $\hat{x}(e) = 0.5$, and $\hat{x}(d) = 1$. The proof from Álvarez-Miranda et al. [2013b] claims that we can follow $\bar{x}(v) = \hat{x}(v)$, $v \in V$, by this definition of the variables. However, this claim is not true because of $0.5 = \bar{x}(d) \neq \hat{x}(d) = 1$, and therefore, no solution can be constructed from the solution $\bar{x}$ to the RNCut model.

In summary, and somewhat broadly speaking, the weaker LP relaxation can be explained as follows. The RNCut formulation can be interpreted as a multi-commodity flow problem in an enlarged graph. However, enlarging the graph opens new possibilities for what is sometimes called *rejoining of flows* Polzin and Daneshmand [2001a]: Flows for different commodities enter a node on different arcs, but leave on the same arc. Such a rejoining can lead to an increased integrality gap.