RESEARCH ARTICLE

Optimal connected subgraphs: Integer programming formulations and polyhedra

Daniel Rehfeldt1,2 | Henriette Franz1,2 | Thorsten Koch1,2

1 Applied Algorithmic Intelligence Methods, Zuse Institute Berlin, Berlin, Germany
2 Chair of Software and Algorithms for Discrete Optimization, TU Berlin, Berlin, Germany

Correspondence
Daniel Rehfeldt, Applied Algorithmic Intelligence Methods, Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany.
Email: rehfeldt@zib.de

Funding information
Federal Ministry of Education and Research, Grant/Award Number: 05M14ZAM; DFG Cluster of Excellence MATH+.

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. In other words, it is not relevant which edges are selected to obtain connectivity. This article is concerned with the exact solution of such problems via integer programming. We analyze and compare (mixed) integer programming formulations with respect to the strength of their linear programming 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.

KEYWORDS
connected subgraph polytope, connected subgraphs, exact solution, induced connectivity, integrality, maximum-weight connected subgraph problem, Steiner tree problem

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. In other words, 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 an induced connectivity span a diverse set of areas: Computational biology [13], wildlife conservation [11], computer vision [9], social network analysis [30], political districting [19], wireless sensor network design [6], and even robotics [4]. Besides this practical relevance, connectivity is also a central and well-studied theoretical concept.

From an optimization perspective, a fundamental prototype for induced connectivity problems is the maximum-weight connected subgraph problem (MWCSP); see, for example, [1]. 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 [2]. 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 [16] 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 [1, 27, 33]. See also the recent survey on solving Steiner tree and related problems by [29].

Various articles discuss theoretical aspects of optimization problems based on induced connectivity, such as the strength of (mixed) integer-programming formulations, for example, [2, 8], polyhedral descriptions, for example, [5, 34], or complexity, for example, [1, 7].

1.1 | Contribution

This article analyzes integer programming (IP) and mixed-integer programming (MIP) 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 linear programming (LP) relaxations. Furthermore, we provide conditions for LP-relaxations of both the MWCSP and the Steiner tree problem to be tight. Along the way, several polyhedral results are given, including a (compact extended) description of the connected subgraph polytope for all graphs with less than four independent vertices.

This article is clustered around the MWCSP, the prototype for optimization problems based on induced connectivity. However, it will also be shown how to apply the results to other induced connectivity problems, such as the (unweighted) Steiner tree problem.

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

$$E[U] := \{ (v,w) \in E | v,w \in U \}.$$ 

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

For $U \subseteq V$ define $\delta(U) := \{ (u,v) \in E | u \in U, v \in V \setminus U \}$ and for a subgraph $G' \subseteq G$ and $U' \subseteq V(G')$ define $\delta_G(U') := \{ (u,v) \in E(G') | u \in U', v \in V(G') \setminus U' \}$. A corresponding notation is used for directed graphs $(V,A)$: For $U \subseteq V$ define $\delta^+(U) := \{ (u,v) \in A | u \in U, v \in V \setminus U \}$ 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

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

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

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

We denote by $\alpha(G)$ the maximum number of independent vertices in graph $G$. Let $(V,A)$ be a directed graph, let $r,t \in V$, and consider an $r-t$ flow $f$ in $(V,A)$. We denote the net flow value of $f$ by $|f| := f(\delta^+(r)) - f(\delta^-(r))$. Given an MWCSP instance $(V,E,p)$ we define $T_p := \{ v \in V | p(v) > 0 \}$.

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 $C(v,w)$. Note that $C(v,w) = \emptyset$ 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 $P_{LP}(F)$, 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 $\mathcal{N}P$-hard; see, for example, [23]. It is even $\mathcal{N}P$-hard to approximate the MWCSP within any constant factor as shown in [1]. 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.
1.3.1 Rooted MWCSP
A close relative of the MWCSP is the rooted maximum-weight connected subgraph problem (RMWCSP); see, for example, [2], 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 $t \in T_f$.

1.3.2 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 in graphs (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 can be formulated as an RMWCSP by setting $T_f := T$ and assigning each nonterminal vertex a weight of $-1$. Many of the hardest Steiner tree benchmark instances are unweighted; see [25] for an overview. Moreover, many theoretical articles consider just the unweighted case; see, for example, [17], who describe an exact polynomial-space algorithm for the USPG.

1.3.3 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 $c : A \rightarrow \mathbb{R}_{\geq 0}$, 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 predefined, 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 instance $(V, A, T, r, c)$. 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 [35] (i.e., one in the original variable space) can thereupon be stated as:

**Formulation 1. Directed cut formulation (DCut)**

$$\min c^T y,$$

s.t. $y(\delta^-(U)) \geq 1$, for all $U \subseteq V$, $r \not\in U$, $U \cap T \neq \emptyset$, (2)

$y(a) \in \{0, 1\}$, for all $a \in A$. (3)

One verifies that the constraints (2) ensure the existence of (directed) paths from the root to each terminal in a feasible solution. We note that a feasible but not optimal solution to DCut is not necessarily the incidence vector of a Steiner arborescence. Indeed, the convex hull of all $y \in \mathbb{N}_0^A$ that satisfy (2) is of blocking type, that is, its recession cone equals $\mathbb{R}_+^A$.

Another well-known formulation, see, for example, [35], is based on flows. This formulation affiliates with each terminal $t \in T \setminus \{r\}$ an $r-t$ flow $f'$.

**Formulation 2. Directed multicommodity flow formulation (DF)**

$$\min c^T y,$$

s.t. $f'(\delta^-(v)) - f'(\delta^+(v)) = \begin{cases} 1 & \text{if } v = t; \\ 0 & \text{if } v \in V \setminus \{r, t\}, \end{cases}$ for all $v \in V, t \in T \setminus \{r\}$, (5)

$f' \leq y$, for all $t \in T \setminus \{r\}$, (6)

$f' \geq 0$, for all $t \in T \setminus \{r\}$, (7)

$y \in \{0, 1\}^A$. (8)
By using the max-flow min cut theorem, one shows that $DF$ is an extended formulation of $DCut$, that is, $\text{proj}_y(\mathcal{P}_{LP}(DF)) = \mathcal{P}_{LP}(DCut)$; see, for example, [14]. Both formulations can be strengthened by the so called flow-balance constraints from [24]:
\[ y(\delta^-(v)) \leq y(\delta^+(v)), \quad \text{for all } v \in V \setminus T. \] (9)

We will refer to the extensions 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 $v_{LP}(DCut_{FB}) = v(DF_{FB})$.

**Proof.** For the case of $|T| = 1$ and $|T| = 2$, the lemma holds already without the flow-balance constraints: The case $|T| = 1$ is clear, and the case $|T| = 2$ (corresponding to the shortest-path problem with non-negative weights) results in a totally unimodular constraint matrix. 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 [3]:
\[
\begin{align*}
  f(\delta^-(v)) - f(\delta^+(v)) &\geq \begin{cases} 
    -1 & \text{if } v = r; \\
    0 & \text{otherwise;}
  \end{cases} \\
  (f + f^t)(\delta^-(v)) - (f + f^t)(\delta^+(v)) &\geq \begin{cases} 
    1 & \text{if } v = r; \\
    -1 & \text{if } v = t; \\
    0 & \text{otherwise;}
  \end{cases} \\
  (f + f^u)(\delta^-(v)) - (f + f^u)(\delta^+(v)) &\geq \begin{cases} 
    1 & \text{if } v = u; \\
    -1 & \text{if } v = r; \\
    0 & \text{otherwise;}
  \end{cases}
\end{align*}
\]

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 $s$, as well as a $s-t$ and a $s-u$ path. Note that any of these paths can be a single node. The flow from $r$ to the terminals $t$ and $u$ is split into a common part $f$, and two separate parts $f^t$ and $f^u$.

Let $(f^t,f^u,y)$ be an optimal LP solution to $DF_{FB}$. Assume that this solution is minimal, that is, for any feasible solution $(\hat{f}^t,\hat{f}^u,\hat{y}) \leq (f^t,f^u,y)$ it holds that $(\hat{f}^t,\hat{f}^u,\hat{y}) = (f^t,f^u,y)$. 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 $a \in A$: 
\[
\hat{f}(a) := \min\{f^t(a), f^u(a)\},
\]
\[
\hat{f}^t(a) := \max\{f^t(a) - f^u(a), 0\},
\]
\[
\hat{f}^u(a) := \max\{f^u(a) - f^t(a), 0\}.
\]

First, we show (10). Let $v \in V$. Because of the assumed minimality of $(f^t,f^u,y)$, we have that $y = \max\{f^t,f^u\}$. Together with (15) we get:
\[
\hat{f}(\delta^-(v)) - \hat{f}(\delta^+(v)) = (f^t + f^u - y)(\delta^-(v)) - (f^t + f^u - y)(\delta^+(v))
\]
\[
= (f^t + f^u)(\delta^-(v)) - (f^t + f^u)(\delta^+(v)) + y(\delta^+(v)) - y(\delta^-(v)).
\]

If $v = r$, then
\[
(f^t + f^u)(\delta^-(v)) - (f^t + f^u)(\delta^+(v)) = 0
\]
and
\[
y(\delta^+(v)) - y(\delta^-(v)) \geq 1,
\]
thus (19) implies that (10) holds. If $v \in \{t,u\}$, then
\[
(f^t + f^u)(\delta^-(v)) - (f^t + f^u)(\delta^+(v)) = 1
\]
and
\[
y(\delta^+(v)) - y(\delta^-(v)) \geq 1.
\]

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
\[ (\hat{f} + \hat{f}')(\delta^-(v)) - (\hat{f} + \hat{f}')(\delta^+(v)) = f'(\delta^-(v)) - f'(\delta^+(v)), \]
which implies (11). Likewise, (13) follows from the definition of $\hat{f}$, $\hat{f}'$, and $\hat{f}''$.

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, for example, [28, 31].

### 2.2 Rooted maximum-weight connected subgraphs

This section discusses the directed variant of the RMWCSP, see [2]: 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 $T_f$ 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, for example, [2], which is based on a well-known formulation for SAP, see, for example, [21]. 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)

\[
\begin{align*}
\max & \quad p^T x, \\
\text{s.t.} & \quad y(\delta^-(v)) = x(v), \quad \text{for all } v \in V \setminus \{r\}, \\
& \quad y(\delta^-(U)) \geq x(v), \quad \text{for all } U \subseteq V \setminus \{r\}, \forall v \in U, \\
& \quad x(t) = 1, \quad \text{for all } t \in T_f, \\
& \quad x \in \{0, 1\}^V, \\
& \quad y \in \{0, 1\}^A.
\end{align*}
\]

Constraints (26) establish the relation between the arc variables and (the actually redundant) vertex variables. Constraints (27) correspond to constraints (2) in the DCut formulation. The constraints make sure that in a feasible solution $S$, for any $v \in V(S)$ there is an $r - v$ path in $S$ as well. Finally, constraints (28) assure that all fixed terminals are contained in any feasible solution.

In [2], 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 [18].

**Formulation 4.** Rooted node separator formulation (RNCut)

\[
\begin{align*}
\max & \quad p^T x, \\
\text{s.t.} & \quad x(C) \geq x(v), \quad \text{for all } v \in V \setminus \{r\}, \forall C \in C(r, v), \\
& \quad x(v) = 1, \quad \text{for all } v \in T_f, \\
& \quad x \in \{0, 1\}^V.
\end{align*}
\]

Constraints (32) ensure that connectivity is fulfilled: By enforcing that for any solution vertex $v$, all $(r, v)$-separators contain at least one solution vertex as well. Constraints (33) ensure the inclusion of all fixed terminals.

Besides the two IP models introduced above, several other formulations for RMWCSP (sometimes including a budget constraint) have been introduced in the literature; see, for example, [2, 11]. However, one can show that these formulations are weaker with respect to the LP-relaxation than both of the above models, see [2] for some such results. Another example is the formulation from [10] that is based on single-commodity flows. However, also this formulation can be shown to be weaker than Formulation 3 by using max-flow/min-cut arguments—similar to corresponding results for minimum spanning tree or Steiner tree problems, which can be found for example in [22].

In [2], 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 B gives a counterexample—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 \( \text{proj}_i(P_{LP}(RSA)) \subseteq P_{LP}(RNCut) \) and the inclusion can be strict.

Proof. The inclusion is essentially shown in [2]. For the sake of completeness, we nevertheless prove it the following. Let \((x, y) \in P_{LP}(RSA)\). We will show that \(x \in P_{LP}(RNCut)\). Consider any \(v \in V \setminus \{r^+ \cup N^+(r)\} \) and a non-empty \(C \subseteq C(r, v)\). Let \(V'\) be the vertices that are reachable from \(r\) in the graph \((V \setminus C, E[V \setminus C])\). We obtain:

\[
x(v) \leq y(\delta^+(V')) \leq y(\delta^-(C)) \leq \sum_{u \in C} y(\delta^-(u)) = x(C),
\]

which shows that (32) is satisfied. Thus, \(x \in P_{LP}(RNCut)\).

Finally, an RMWCSP instance for which \(\text{proj}_i(P_{LP}(RSA)) \not\subseteq P_{LP}(RNCut)\) holds is given in Figure B1 in Appendix B. The point \(x \in \mathbb{R}^V\) defined by \(x(r) := x(c) := x(e) := 1\), and \(x(a) := x(b) := x(d) := 0.5\) is contained in \(P_{LP}(RNCut)\). However, \(x \not\in \text{proj}_i(P_{LP}(RSA))\): Consider any \(y \in \mathbb{R}_{\geq 0}^V\) with \(x(v) = y(\delta^-(v))\) for all vertices \(v\). Because of \(x(d) = 0.5\), we have that either \(y((a, d)) < 0.5\) or \(y((b, d)) < 0.5\) holds. Thus, we have either \(y(\delta^-(\{a, d, c\})) < 1 = x(c)\) or \(y(\delta^-(\{b, d, e\})) < 1 = x(e)\), which contradicts (27). In particular, it holds that \(p(x) = v_{LP}(RNCut) = -1.5\), but \(v_{LP}(RSA) = -2\).

One can strengthen the RSA formulation by inequalities similar to the flow-balance constraints. However, these constraints depend on the objective vector (i.e., they are only valid for specific node-weight assignments), so they cannot directly be used for polyhedral results.

\[
y(\delta^-(v)) \leq y(\delta^+(v)) \quad \text{for all } v \in V \setminus (T_r \cup T_p). \tag{35}
\]

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 [33] 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 \((v, z)\) 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 [16]. Note that in [16], 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)**

\[
\begin{align*}
\min \quad & x(V) - 1, \\
\text{s.t.} \quad & x(C) \geq 1, \quad \text{for all } t, u \in T, t \neq u, C \subseteq C(t, u), \tag{38} \\
& x(v) = 1, \quad \text{for all } v \in T, \tag{39} \\
& x(v) \in \{0, 1\}, \quad \text{for all } v \in V, \tag{40} \\
& x(v) \in \{0, 1\}, \quad \text{for all } v \in V. \tag{41}
\end{align*}
\]

Second, we look at the well-known bidirected cut formulation (BCut) 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. This property will be exploited in (the subsequent) Theorem 6 to reduce the size of instances with a small number of independent vertices.

**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

\[
v_{LP}(BDCut(I)) = v_{LP}(BDCut(I')) + 1.
\]

**Proof.** Throughout the proof, we assume that \(u\) is the root for the \(BDCut\) formulation, that is, \(r = u\). It is well known that the choice of the root does not affect \(v_{LP}(BDCut)\); see, for example, [21] (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 \(I'\) be the new vertex. So, \(V' = (V \setminus \{r, t\}) \cup \{r'\}\).

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 [31]. Create an optimal solution \(\tilde{y}\) (which can possibly be equal to \(y\)) 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 \(a \in \delta^-(t) \setminus \{(r, t)\}\). For any \(a = (r', v) \in \delta^+_U(r')\) 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 \(y'(a) := \tilde{y}((t, v))\); 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 \(\tilde{y}\) to \(BDCut(I)\). First, \(\tilde{y}((r, t)) := 1\), \(\tilde{y}((t, r)) := 0\). Second, \(\tilde{y}(a) := y'(a)\) for all \(a \in A' \cap A\), and \(\tilde{y}(a) := 0\) for all \(a \in \delta^-(\{r, t\})\). Next, consider the remaining edges \(\delta^+((r, t))\). If \((r, v), (t, v) \in A\) set \(\tilde{y}((r, v)) := y'((r', v))\), \(\tilde{y}((t, v)) := 0\); otherwise, for \(a = (r, v)\) or \(a = (t, v)\) set \(\tilde{y}(a) = y'((r', v))\).

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

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

**Proof.** Consider a USPG instance \(I = (G, T, c)\) with \(a(G) \leq 3\). 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 \(a(G) \leq 3\) 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

\[
y(\delta^+(v)) < y(\delta^-(v)).
\]

Because of \(a(G) \leq 3\), 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 \(a \in T' \setminus \{r\}\) is adjacent to \(v\) and let \(f'\) be a unit flow from \(r\) to \(t\) such that \(f' \leq y\); define \(\tilde{y}((q, u)) := y((q, u)) - f'(q, u) + f'(u, q)\) 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 \((\text{remaining})\) \(a \in A' \setminus \delta^-(v)\) 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 an \(U \subseteq V \setminus \{r\}\) with \(U \cap T' \neq \emptyset\) and \(y'(\delta^-(U)) < 1\). Because
Comparison of edge and node-based formulation

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

\[
y(\delta^-(\tilde{U})) = y'(\delta^-(\tilde{U})) = y'((r, v)) - y'(\delta^+(v)) < 1,
\]

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; that is, there exist USPG instances such that \( a(G) = 4 \) and \( v_{LP}(BDCut) \neq v(BDCut) \); see, for example, [14, 15].

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 [12]. 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 bidirectional cut formulation. As the next proposition shows, BDCut is always stronger than TNCut and the relative gap can be quite large.

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

\[
\sup \left\{ \frac{v_{LP}(BDCut)}{v_{LP}(TNCut)} \right\} \geq 2,
\]

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 [31]. Let \( t, u \in T \) with \( t \neq u \) and \( C_t \in C(t, u) \). We will show that \( C(t, u) \) satisfies (39). If \( C_t \cap T = \emptyset \), then \( x(C_t) \geq 1 \), because \( x(q) \geq 1 \) for all \( q \in T \) due to (2) and the definition of \( x \). Thus, (39) holds. If \( C_t \cap T \neq \emptyset \), let \( U_t \) be the connected component in the graph induced by \( V \setminus C_t \) with \( r \in U_t \). By definition of \( C_t \), either \( t \notin U_t \) or \( u \notin U_t \). Therefore, \( y(\delta^+(U_t)) \geq 1 \), which implies \( y(\delta^+(C_t)) \geq 1 \) because of \( \delta^+(U_t) \subseteq \delta^-(C_t) \). Now we obtain from the definition of \( x \) that

\[
x(C_t) \geq y(\delta^- (C_t)) \geq 1.
\]

Finally, by construction of \( x \) we have that

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

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 \( k + k^2 \) nodes, \( k + k^2 \) edges, and \( k \) terminals defined as follows. Let \( t_i \) for \( i = 1, \ldots, k \) be the terminals and define for each \( i \in \{1, \ldots, k\} \) Steiner nodes \( v_{ij}, j = 1, \ldots, k \). For each \( i \in \{1, \ldots, k\} \) define edges \( \{ t_i, v_{ik} \}, \{ t_{i+1} \text{mod} k, v_{ik}\} \), and \( \{ v_{ij}, v_{i+1j}\} \) for \( j = 1, \ldots, k - 1 \). Instance \( I_1 \) 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 \( k^2/2 + k - 1 \). On the other hand, it holds that \( v_{LP}(BDCut(I_k)) = v(BDCut(I_k)) \), because \( I_k \) consists of a cycle (and is thus in particular series-parallel); see, for example, [20]. Any optimal Steiner tree in \( I_k \) contains all edges except for those between two
adjacent terminals \( t_i \) and \( t_{i+1} \). Thus, \( v_{LP}(BDCut(I_k)) = v(BDCut(I_k)) = (k+1)(k-1) = k^2 - 1 \). Consequently,

\[
\lim_{k \to \infty} \frac{v_{LP}(DCut(I_k))}{v_{LP}(TCut(I_k))} = \lim_{k \to \infty} \frac{k^2 - 1}{k^2 + k - 1} = 2.
\]

(45)

which concludes the proof.

\[\blacksquare\]

**Corollary 8.** The (relative) integrality gap of \( TCut \) is at least 2.

Note that one can strengthen \( TCut \) by constraints that correspond to the flow-balance constraints for \( BDCut \); see [16]. However, if compared to \( BDCut_{FB} \), the results of Proposition 7 remain the same for this stronger version of \( TCut \).

### 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 the MWCSP that use only node variables. The probably best known one, see, for example, [34], is given below.

**Formulation 6.** Node separator formulation (\( NCut \))

\[
\begin{align*}
\text{max} & \quad p^T x, \\
\text{s.t.} & \quad x(v) + x(w) - x(C) \leq 1, \quad \text{for all } v, w \in V, v \neq w, \ C \in C(v, w), \\
& \quad x(v) \in \{0, 1\}, \quad \text{for all } v \in V.
\end{align*}
\]

Constraints (47) guarantee that in a feasible solution \( S \) for any disjoint \( v, w \in V(S) \), at least one node of each \( (v, w) \)-separator is also contained in \( S \).

In this section, we are interested in cases where \( NCut \) has no integrality gap. Recalling the invariance of the \( BDCut \) integrality gap to the contraction of terminals, one might wonder whether a corresponding property holds for the MWCSP and the \( NCut \) formulation. As show in the next proposition, the answer is yes. As before, we will exploit this property for showing an integrality condition based on the independence number: By exhaustively contracting all adjacent, positive vertices. 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 \( I \) by contracting \( \{t, u\} \) into a new vertex \( t' \). It holds that \( v_{LP}(NCut(I')) \leq v_{LP}(NCut(I)) \), because any \( x' \in P_{LP}(NCut(I')) \) can be mapped to a \( x \in P_{LP}(NCut(I)) \) with \( p^T x = p'^T x' \) defined by \( x(v) := x'(v) \) for all \( v \in V \cap V' \), and \( x(t) := x(u) := x'(t') \).

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

\[
x(t) = x(u).
\]

(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 \( x' \in P_{LP}(NCut(I')) \). Suppose this is not the case. Then there are \( a, b \in V' \) and an a-b separator \( C_{ab}' \subset V' \) such that

\[
x'(a) + x'(b) - x'\left(C_{ab}'\right) > 1.
\]

(50)

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

\[
x(a) + x(b) - x(t) = x'(a) + x'(b) - x'(t') > 1,
\]

(51)

and therefore

\[
\min\{x(a), x(b)\} > x(t).
\]

(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 \( C_{qs} \in C(s, t) \) such that
\[
x(t) + x(q) - x(C_{qs}) = 1.
\]
(53)

Similarly, there is a \( s \in V \setminus \{ t, u \} \) and a \( C_{su} \in C(s, u) \) with \( x(u) + x(s) - x(C_{su}) = 1 \). At least one such combination \( q, C_{qs} \), or \( s, C_{su} \) satisfies \( u \notin C_{qs} \) or \( t \notin C_{su} \), otherwise we could increase \( x(u) \) and \( x(t) \). Assume w.l.o.g. \( u \notin C_{qs} \). From (53), we obtain
\[
x(C_{qs}) \leq \min \{ x(t), x(q) \}.
\]
(54)

Thus, (53) and (52) imply \( a, b \notin C_{qs} \). We note that \( C_{qs} \notin C(a, q) \), because (52) and (53) imply
\[
x(a) + x(q) - x(C_{qs}) > 1.
\]
(55)

Likewise, \( C_{qs} \notin C(b, q) \). Consequently, any path from \( \{ t, u \} \) to \( a \) or \( b \) needs to cross \( C_{qs} \); otherwise, the latter would not separate \( q \) and \( t \). Therefore, \( \tilde{C}_{ab} := (C_{ab} \setminus \{ t' \}) \cup C_{qs} \) separates \( a \) and \( b \) (in the original graph). However, from (50) and (54) we obtain
\[
1 < x(a) + x(b) - x'(C_{ab}) \leq x(a) + x(b) - x(\tilde{C}_{ab}),
\]
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 new 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 \( k(e) := \infty \). By the max-flow/min-cut theorem there is an \( a \)-\( b \) flow \( f \) with \( |f| = x(b) \) in this extended network. Define the directed MWCSP \( I_r := ((V, A), T_{r}, r, p) \) with \( T_{r} := \{ a \} \) and \( r := a \), and set \( y := f |_{A} \). Because of the optimality and minimality of \( x \) it holds that \( (x, y) \in P_{LP}(RSA(I_r)) \). Thus, \( v_{LP}(NCut(I_r)) \leq v_{LP}(RSA(I_r)) \). Furthermore, \( y \) satisfies constraints (35). Because of \( v(NCut(I_r)) = v(RSA(I_r)) \), Lemma 3 implies that \( v_{LP}(NCut(I_r)) = v(NCut(I)) \).

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 [34].

**Theorem 11.** If \( a(G) \leq 2 \), then \( P_{LP}(NCut) \) is integral.

**Proof.** Let \( p \in \mathbb{R}^V \). If \( a(G) \leq 2 \), 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, \( P_{LP}(NCut) \) is integral.

[34] also shows that \( P_{LP}(NCut) \) is integral only if \( a(G) \leq 2 \).

\[\text{Figure 2: MWCSP instance with given node weights}\]
3.1.1 Indegree constraints

Given an undirected graph \( G = (V, E) \), a \( d \in \mathbb{Z}^n \) is an in-degree 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 in-degree vector \( d \) the corresponding in-degree inequality is given as

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

where \( x \in \mathbb{R}^V \) are the node variables. \cite{26} shows that the in-degree inequalities describe the connected subgraph polytope if \( G \) is a tree. Furthermore, \cite{34} shows conditions for (57) to be facet inducing, and show that the constraints can be separated in linear time. It is further shown that the constraints (57) can strengthen the NCut formulation.

3.2 Edge-based formulations

An edge-based formulation for the directed MWCSP is introduced in \cite{1}, 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 graph \( (V, A_r) \) of the given undirected graph. Let \((V_r, A_r)\) be the directed graph defined as follows with an additional node \( r: V_r := V \cup \{r\}, A_r := A \cup \{(r, v) | v \in V\} \). Define the following extended MWCSP formulation based on the new graph \((V, E, A_r)\).

**Formulation 7.** Extended Steiner arborescence formulation (ESA)

\[
\begin{align*}
\max & \quad p^T x, \\
\text{s.t.} & \quad y(\delta^-(v)) = x(v), \quad \text{for all } v \in V, \tag{58} \\
& \quad y(\delta^-(U)) \geq x(v), \quad \text{for all } U \subseteq V, v \in U, \tag{59} \\
& \quad y(\delta^+(r)) \leq 1, \tag{60} \\
& \quad x \in \{0, 1\}^V, \tag{61} \\
& \quad y \in \{0, 1\}^A_r. \tag{62}
\end{align*}
\]

ESA is almost the same as RSA (Formulation 3). The additional constraint (61) ensures that at most one arc incident to the (artificial) root node \( r \) is selected. Otherwise, a solution could consist of several connected components in the original graph \((V, E)\).

The remainder of this section aims to prove an integrality condition for \( \text{proj}_x(P_{\text{LP}}(ESA)) \) based on the independence number. Our approach can be divided in two parts. First, we show that for any MWCSP instance with \( 1 \leq |T_p| \leq 3 \) there is an optimal LP solution \((x, y)\) with \( x(v) = 1 \) for a \( v \in T_p \). In the second part (following Lemma 15), we use this \( v \) as a root node and apply the same principal ideas already used in Section 2.3.1 for the USPG: We show the invariance of the integrality gap under edge contraction and reduce any MWCSP instance with bounded independence number to an MWCSP instance with bounded number of positive vertices. We start with an easy technical result.

**Lemma 12.** Let \((x, y)\) be an optimal LP solution \((x, y)\) to ESA, and let \( v \in V \). There is a \( y \in \mathbb{R}^A_r \) with \( y((r, v)) = x(v) \) such that \((x, y)\) 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 \( f_k \leq y \) with \( |f_k| = \kappa \) and \( f_k((r, v)) = 0 \). Define a new solution \((x, \tilde{y})\) with

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

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

\[
\tilde{y}(\delta^-(u)) = y(\delta^-(u)) + f_k(\delta_D^+(u)) - f_k(\delta_D^-(u)) - f_k((r, u)) = y(\delta^-(u)) = x(u).
\]

Similarly, because of \( \tilde{y}((r, v)) = y((r, v)) + \kappa \) and \( f_k(\delta_D^-(v)) + f_k(\delta_D^+(v)) = \kappa \) 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 \( f_k(\delta_D^-(U)) + f_k(\delta_D^+(U)) = \kappa \) we obtain \( \tilde{y}(\delta^-(U)) = y(\delta^-(U)) \). Second, assume \( v \notin U \). In this case, flow conservation of \( f_k \) implies that

\[
\tilde{y}(\delta^-(U)) = y(\delta^-(U)) + f_k(\delta^+(U)) - f_k(\delta^-(U)) = y(\delta^-(U)).
\]

\[\]
Note that for (exact) optimization, ESA only requires arcs from $r$ to vertices in $T_p$, that is, we can restrict ourselves to the arc set $A^+_r := A \cup \{(r,t) \mid t \in T_p\}$. 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, defined on the graph $(V, A^+_r)$, as $ESA^+$. In practice, it is advisable to add additional $|T_p|$ symmetry-breaking constraints similar to those from [16] to $ESA^+$. As to the LP-approximation of $ESA^+$, one obtains the following result.

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

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

(64)

In this proof, we only consider minimal optimal LP solutions, that is, solutions for which no entry can be reduced without losing 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 an $r-t$ flow $f'$ with $f' \leq y'$ for $t \in T_p$ with $|f'| = y'((r,v))$. 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 [21] for the SPG bidirected cut formulation. Suppose there is a $U \subseteq V$ and a $u \in U$ with

$$x'(u) > y'(\delta^-(U)).$$

(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

$$y'(\delta^-(W)) = x'(t).$$

(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

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

With (66), it follows that $y'((\delta^-(U)) \geq y'(\delta^-(U \cup W))$, which leads to a contradiction. □

**Corollary 14.** $v_{LP}(ESA) = v_{LP}(ESA^+)$. Further, we require the following result. The (quite lengthy) proof is given in the Appendix.

**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 $t \in T_p$.

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 $\text{proj}_{\mathcal{I}}(P_{LP}(ESA))$ 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 Lemmas 12 and 15, 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 RMWCS $I'_{\delta} = (V', E', T'_p, p')$ with $T'_p := \{t\}$. For simplicity, we deviate from the assumption...
that fixed terminals have 0 weight. It holds that \( v(ESA(I')) = v(RSA(I'_0)) \) and \( v_{LP}(ESA(I')) = v_{LP}(RSA(I'_0)) \). We will show that

\[
v_{LP}(RSA(I'_0)) = v(RSA(I'_0)), \tag{67}
\]

which concludes the proof. Let \((x, y)\) be the restriction of \((\bar{x}, \bar{y})\) to \((V', A')\). Note that \((x, y)\) is an optimal LP solution to \(RSA(I'_0)\). Suppose that \((67)\) does not hold. Thus, by Lemma 3 there is a \(v \in V' \setminus (T'_p \cup T'_f)\) with

\[
y(\delta^+(v)) < y(\delta^-(v)). \tag{68}
\]

The case \(|T'_p| < 3\) can be readily ruled out by a flow argument. So assume \(|T'_p| = 3\). Because of \(a(G) \leq 3\), 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 \(I'_u = (V', E', T'_p, p')\) with \(T'_p := \{t, u\}\). Further, construct an optimal 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 \(I'_u\) it holds that \(p'(v) < 0\) (otherwise, \(v\) would have been contracted into \(u\)). Thus, \(p'^Tx' > p^Tx\). 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 \(\text{proj}_x(P_{LP}(ESA))\) 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 [1] states that the directed equivalents of ESA and (a slight generalization of) NCut induce the same polyhedral relaxation of the directed connected subgraph polytope. This result suggests that the same relation holds for the undirected case. Unfortunately, the result from [1] is not correct (the proof suffers from a similar problem as that discussed in Appendix B 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:

\[
\text{proj}_x(P_{LP}(ESA)) \subset P_{LP}(NCut). \tag{69}
\]

**Proof.** Let \((x, y) \in P_{LP}(ESA)\) and let \(a, b \in V, a \neq b\). Let \(C \subseteq C(a,b)\) 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

\[
y(\delta^-(\bar{U}_a)) + y(\delta^-(\bar{U}_b)) = y(\delta^-(\bar{U}_a \cup \bar{U}_b)) + y(\delta^-(C)), \tag{70}
\]

where we use \(\delta^- := \delta^-_{D,1}\). Thus,

\[
x(a) + x(b) \leq y(\delta^-(\bar{U}_a)) + y(\delta^-(\bar{U}_b)) \leq y(\delta^+(r)) + y(\delta^-(C)) \leq 1 + x(C). \tag{73}
\]

An example for a strict inclusion is given in Figure 2. Consider the following point that is in \(P_{LP}(NCut)\), but not in \(\text{proj}_x(P_{LP}(ESA))\): Set the \(x\) values of all negative weight node variables to 0.5 and the remainder to 1. To see that this point is indeed not in \(\text{proj}_x(P_{LP}(ESA))\), consider arc variables \(y\) such that \(y(\delta^-(v)) = x(v)\) for all vertices \(v\). Because of \(x(d) = 1\), we can proceed as in Lemma 12 and assume that \(y((r, d)) = 1\). First, we have \(y(\delta^-((a, b, c)) \geq x(a) = 1\). Because of \(x(b) = x(c) = 0.5\), this inequality implies that \(y((d, b)) = y((f, c)) = 0.5\). Second, we have \(y(\delta^-((c, e, f))) \geq x(f) = 1\). Because of \(x(c) = 0.5\), we have that \(y((d, e)) \leq 0.5\). Thus, together with \(y((f, c)) = 0.5\) we obtain that \(y(\delta^-(c)) \geq 1\). This implies that \((x, y) \notin P_{LP}(ESA)\), because \(y(\delta^-(c)) = x(c) = 0.5\) would need to hold otherwise.

Next, we consider the indegree constraints. Following [34], we define

\[
Q' := \{x \in \mathbb{R}^V_{\geq 0} | x \text{ satisfies all indegree constraints}\}. \tag{74}
\]

While \(Q' \not\subset P_{LP}(NCut)\), see, for example, [34], the indegree constraints cannot improve the ESA formulation, as the following proposition shows.
Proposition 19. The following inclusion holds and can be strict:

\[
\text{proj}_{x}(P_{LP}(ESA)) \subset Q'.
\] (75)

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 \((x, y) \in P_{LP}(ESA)\). First, note that constraints (59) and (60) imply for all \( \{v, w\} \in E \) that

\[
\min\{y(\gamma_{D_r}(v)), y(\gamma_{D_r}(w))\} \geq y((v, w)) + y((w, v)).
\] (76)

Let \( d \) be an indegree vector. It holds that

\[
\sum_{v \in V} x(v) = \sum_{a \in A_r} y(a) \\
\leq \sum_{a \in A} y(a) + 1 \\
= \sum_{\{v, w\} \in E} (y((v, w)) + y((w, v))) + 1 \\
\overset{(76)}{\leq} \sum_{\{v, w\} \in E} \min\{y(\gamma_{D_r}(v)), y(\gamma_{D_r}(w))\} + 1 \\
\leq \sum_{v \in V} d_v x(v) + 1,
\]

which implies that (57) is satisfied by \( x \); thus, \( x \in Q' \). For a strict inclusion consider the graph in Figure 2 and the point \( x \) as defined in the proof of Proposition 18. To see that this point satisfies all indegree constraints, consider an indegree vector \( q \) that minimizes \( \sum_{v \in V} q_v x(v) \). Because all adjacent vertices of \( a, d, \) and \( f \) have value 0.5, we obtain \( q_a = q_d = q_f = 0 \). Thus, \( \sum_{v \in \{b, c, e\}} q_v x(v) = 3.5 \). Consequently, \( \sum_{v \in V} q_v x(v) + 1 = 4.5 = \sum_{v \in V} x(v) \), which shows that the indegree constraint is satisfied. \( \blacksquare \)

Summarizing the results of this section, one obtains:

**Theorem 20.** It holds that

\[
\text{proj}_{x}(P_{LP}(ESA)) \subset Q' \cap P_{LP}(NCut),
\] (77)

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 | CONCLUSION

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.

Finally, we note that the theoretical predominance of edge-based formulations over node-based ones is complemented by recent computational results: In [32], a MWCSP solver that uses the \( ESA_{FB} \) formulation is shown to significantly outperform all other (and in particular node-based) MWCSP solvers from the literature.

ACKNOWLEDGMENTS

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 DFG Cluster of Excellence MATH+. Open Access funding enabled and organized by Projekt DEAL.
REFERENCES

[1] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel, “The maximum weight connected subgraph problem,” Facets of combinatorial optimization, Springer, Berlin, Heidelberg, 2013, pp. 245–270. https://doi.org/10.1007/978-3-642-38189-8_11.

[2] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel, The rooted maximum node-weight connected subgraph problem, Springer, Berlin, Heidelberg, 2013, 300–315. https://doi.org/10.1007/978-3-642-38171-3_20.

[3] M. Ball, W. Liu, and W. Pulleyblank, “Two terminal Steiner tree polyhedra,” Contributions to operations and economics - the twentieth anniversary of CORE, MIT Press, Cambridge, MA, 1989, pp. 251–284.

[4] J. Banfi, Multirobot exploration of communication-restricted environments. Ph.D. thesis, Politecnico di Milano, 2018.

[5] M. D. Biha, H. L. M. Kerivin, and P. H. Ng, Polyhedral study of the connected subgraph problem, Discret. Math. 338 (2015), no. 1, 80–92. https://doi.org/10.1016/j.disc.2014.08.026.

[6] A. Buchanan, J. S. Sung, S. Butenko, and E. L. Pasiliao, An integer programming approach for fault-tolerant connected dominating sets, INFORMS J. Comput. 27 (2015), no. 1, 178–188. https://doi.org/10.1287/ijoc.2014.0619.

[7] A. Buchanan, Wu, and S. Butenko, Algorithms for node-weighted Steiner tree and maximum-weight connected subgraph, Networks 72 (2018), no. 2, 238–248. https://doi.org/10.1002/net.21825.

[8] R. Carvaljal, M. Constantino, M. Goycoolea, J. P. Vielma, and A. Weintraub, Imposing connectivity constraints in forest planning models, Oper. Res. 61 (2013), no. 4, 824–836. https://doi.org/10.1287/opre.2013.1183.

[9] C. Chen and K. Grauman, Efficient activity detection with max-subgraph search, Proc. 2012 IEEE Conf. Comput. Vis. Pattern Recognit., Providence, RI, June 16–21, 2012, pp. 1274–1281. https://doi.org/10.1109/CVPR.2012.6247811.

[10] J. Conrad, C. P. Gomes, J. van Hoeve, A. Sabharwal, and J. Suter, “Contributions to operations and economics - the twentieth anniversary of CORE, MIT Press, Cambridge, MA, 1989, pp. 251–284.

[11] M. X. Goemans, X. Wang, and M. X. Goemans, Polyhedral study of the connected subgraph problem, Discret. Math. 338 (2015), no. 1, 80–92. https://doi.org/10.1016/j.disc.2014.08.026.

[12] M. D. Biha, H. L. M. Kerivin, and P. H. Ng, Polyhedral study of the connected subgraph problem, Discret. Math. 338 (2015), no. 1, 80–92. https://doi.org/10.1016/j.disc.2014.08.026.

[13] I. Ljubić, Thorsten Koch, and A. Martin, “SteinLib: An updated library on Steiner tree problems in graphs,” Steiner trees in industries, D.-Z. Du and X. Cheng (eds.), Kluwer, 2001, Alphen aan den Rijn, Netherlands, pp. 285–325.

[14] S. Butenko, B. Dilkina and C. P. Gomes, “Solving Steiner tree problems in wildlife conservation,” Proc. 7th Int. Conf. Integ. AI OR Techniques in constraint programming for combinatorial optimization problems, Springer, Berlin, Heidelberg, 2007, pp. 16–28.

[15] B. Dilkina and C. P. Gomes, Solving connected subgraph problems in wildlife conservation, Proc. 7th Int. Conf. Integ. AI OR Techniques in constraint programming for combinatorial optimization problems, Springer, Berlin, Heidelberg, 2007, pp. 16–28.

[16] I. Ljubić, Thorsten Koch, and A. Martin, “SteinLib: An updated library on Steiner tree problems in graphs,” Steiner trees in industries, D.-Z. Du and X. Cheng (eds.), Kluwer, 2001, Alphen aan den Rijn, Netherlands, pp. 285–325.

[17] S. Butenko, B. Dilkina and C. P. Gomes, “Solving Steiner tree problems in wildlife conservation,” Proc. 7th Int. Conf. Integ. AI OR Techniques in constraint programming for combinatorial optimization problems, Springer, Berlin, Heidelberg, 2007, pp. 16–28.

[18] I. Ljubić, Thorsten Koch, and A. Martin, “SteinLib: An updated library on Steiner tree problems in graphs,” Steiner trees in industries, D.-Z. Du and X. Cheng (eds.), Kluwer, 2001, Alphen aan den Rijn, Netherlands, pp. 285–325.

[19] S. Butenko, B. Dilkina and C. P. Gomes, “Solving Steiner tree problems in wildlife conservation,” Proc. 7th Int. Conf. Integ. AI OR Techniques in constraint programming for combinatorial optimization problems, Springer, Berlin, Heidelberg, 2007, pp. 16–28.

[20] I. Ljubić, Thorsten Koch, and A. Martin, “SteinLib: An updated library on Steiner tree problems in graphs,” Steiner trees in industries, D.-Z. Du and X. Cheng (eds.), Kluwer, 2001, Alphen aan den Rijn, Netherlands, pp. 285–325.

[21] S. Butenko, B. Dilkina and C. P. Gomes, “Solving Steiner tree problems in wildlife conservation,” Proc. 7th Int. Conf. Integ. AI OR Techniques in constraint programming for combinatorial optimization problems, Springer, Berlin, Heidelberg, 2007, pp. 16–28.
APPENDIX A: PROOF OF LEmma 15

**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 $|T_p| = 1$ 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 $y(\delta^-(a)) = 0$. Thus, also $y(\delta^+(b)) = 0$. 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

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

which implies also $\beta := y(\delta^+(b)) > 0$. Let $\kappa := \frac{1}{\beta}$. Define $\tilde{y} \in \mathbb{R}^{A^+}$ 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 $y(\delta^-(a)) = 0$. 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 $y(\delta^-(a)) = 0$, 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 $f^c_a$ from $r$ to $c$ with $f^c_a = f^c_a((r, b)) = y(r, b)$ and $f^c_a \leq \tilde{y}$—otherwise, we decrease $y(r, b)$ and increase $y(r, a)$. Let $f^c_a$ and $f^c_b$ be maximum flows from $a$ to $b$ and $c$ with $f^c_a \leq \tilde{y}$ and $f^c_b \leq \tilde{y}$. If $|f^c_a| = |f^c_b| = 0$, 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 $|f^c_a| > 0$ or $|f^c_b| > 0$. Thus, $x(a) > 0$. Suppose $x(b) < 1$ and $x(c) < 1$. Note that either $|f^c_a| = 0$, or both $|f^c_a| > 0$ and $|f^c_b| > 0$. First, suppose $|f^c_a| = 0$. Thus, $y(\delta^-(b)) = 0$ and

$$\sum_{v \in V \setminus \{a, b\}} p(v)y(\delta^-(v)) > 0, \quad (A2)$$

Define $\kappa := \frac{x(a)}{|f^c_a|}$. Define $\tilde{y} \in \mathbb{R}^{A^+}$ by

$$\tilde{y}(e) := \max\{y(e), \kappa f^c_a(e)\}, \quad (A3)$$

for all $e \in A^+$, and define $\tilde{x} \in \mathbb{R}^V$ accordingly. We have $\kappa x(c) = \tilde{x}(c)$. Thus, $(A2)$ implies $p^T\tilde{x} > p^T x$.

Second, suppose $|f^c_a| > 0$ and $|f^c_b| > 0$. In this case, $(A1)$ holds. Furthermore, $y(\delta^+(b)) = f^c_a(\delta^+(b))$. Thus, we can proceed as before and multiply both $f^c_a$ and $f^c_b$ 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 $x(a), x(b) \in \{0, 1\}$. 

---

[32] D. Rehfeldt, H. Franz, and T. Koch, *Optimal connected subgraphs: Formulations and algorithms*, Technical report 20–23, ZIB, Takustr, Berlin, 2020.

[33] 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 J. Optim. 29 (2019), no. 1, 369–398. https://doi.org/10.1137/17M1145963.

[34] Y. Wang, A. Buchanan, and S. Butenko, *On imposing connectivity constraints in integer programs*, Math. Program. 166 (2017), 1–31. https://doi.org/10.1007/s10107-017-1117-8.

[35] R. Wong, *A dual ascent approach for Steiner tree problems on a directed graph*, Math. Program. 28 (1984), 271–287. https://doi.org/10.1007/BF02612335.

How to cite this article: D. Rehfeldt, H. Franz, and T. Koch, *Optimal connected subgraphs: Integer programming formulations and polyhedra*, Networks, 80 (2022), 314–332. https://doi.org/10.1002/net.22101
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 \( \overline{f}^a \) be a maximum \( t \)-a flow with \( \overline{f}^a \leq \overline{k} \).

**Claim 2.** It holds that \( |\overline{f}^a| > 0 \).

**Proof.** Suppose \( |\overline{f}^a| = 0 \). Let \( \nabla \subset V \) 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 \( |\overline{f}^a| = 0 \), we have \( a \not\in \nabla \). Since \((x, y)\) is optimal, we have \( b \not\in \nabla \). Because of \( a \not\in \nabla \) and \( x(a) = 1 \), we obtain

\[
 f^a(\delta^+(\nabla)) - f^a(\delta^-(\nabla)) = 1. \tag{A4}
\]

From the definition of \( \nabla \), we get

\[
 f^b(\delta^+(\nabla)) \geq f^b(\delta^+(\nabla)). \tag{A5}
\]

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

\[
 f^b(\delta^+(\nabla)) - f^b(\delta^-(\nabla)) < 1. \tag{A6}
\]

From (A4)–(A6), we get

\[
 f^b(\delta^-(\nabla)) > f^a(\delta^-(\nabla)). \tag{A7}
\]

Thus, there is a \( u \in \nabla \setminus \{t\} \) 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 \( \nabla \), 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 \( |\overline{f}^a| < 1 \). Otherwise the proof would already be complete, since the support graphs of \( f^a \) and \( f^b \) would be arc disjoint. Let \( \overline{f}^a := \overline{f}^a - f^a \). Further, for all \( e \in A \) define \( \overline{f}^b(e) := \max\{0, f^b(e) - f^a(e)\} \). Note that \( \overline{f}^a \) is a flow, but \( \overline{f}^b \) in general not. We further have

\[
 \sum_{v \in V} p(v)(\overline{f}^a + \overline{f}^b)(\delta_D(v)) = \sum_{v \in V \setminus \{t\}} p(v) x(v). \tag{A8}
\]

**Claim 3.** It holds that

\[
 \frac{1}{|\overline{f}^a|} \sum_{v \in V} p(v)(\overline{f}^a + \overline{f}^b)(\delta_D(v)) = \sum_{v \in V \setminus \{t\}} p(v) \overline{f}^a(\delta_D(v)). \tag{A9}
\]

**Proof.** Because \((x, y)\) is optimal and \( f^a = \overline{f}^a + f^a \), we obtain from (A8) that for any sufficiently small \( \epsilon > 0 \):

\[
 \left(1 + \frac{\epsilon}{|\overline{f}^a|}\right) \sum_{v \in V} p(v)(\overline{f}^a + \overline{f}^b)(\delta_D(v)) + \left(1 - \frac{\epsilon}{|\overline{f}^a|}\right) \sum_{v \in V \setminus \{t\}} p(v) \overline{f}^a(\delta_D(v)) \leq \sum_{v \in V \setminus \{t\}} p(v) x(v), \tag{A10}
\]

and

\[
 \left(1 - \frac{\epsilon}{|\overline{f}^a|}\right) \sum_{v \in V} p(v)(\overline{f}^a + \overline{f}^b)(\delta_D(v)) + \left(1 + \frac{\epsilon}{|\overline{f}^a|}\right) \sum_{v \in V \setminus \{t\}} p(v) \overline{f}^a(\delta_D(v)) \leq \sum_{v \in V \setminus \{t\}} p(v) x(v). \tag{A11}
\]

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

\[
 0 \leq \frac{\epsilon}{|\overline{f}^a|} \sum_{v \in V} p(v)(\overline{f}^a + \overline{f}^b)(\delta_D(v)) - \frac{\epsilon}{|\overline{f}^a|} \sum_{v \in V \setminus \{t\}} p(v) \overline{f}^a(\delta_D(v)) \leq 0, \tag{A12}
\]

which proves (A9).
(Proof of Lemma 15 continued.) Set

\[ \kappa := \left( 1 + \frac{\|Î\|^2 + \|Î\|^2}{\|Î\|^2} \right). \]  

(A13)

Because of (A8) and (A9), we obtain

\[ \kappa \sum_{v \in V} p(v)(Î^a + Î^b)(δ_G(v)) = \sum_{v \in V} p(v)x(v). \]  

(A14)

Define \( \Tilde{y} \in \mathbb{R}^{A^*_+} \) by

\[ \Tilde{y}(e) := \max \{ y(e), \kappa Î^a(e), \kappa Î^b(e) \} \]

(A15)

for all \( e \in A^*_+ \), and define \( \Tilde{x} \in \mathbb{R}^V \) accordingly. It holds that \( p^T \Tilde{x} = p^T x \), and \( \Tilde{x}(c) = 1 \).

\[ \blacksquare \]

APPENDIX B: NODE SEPARATORS AND REJOINING OF FLOWS

Consider the directed RMWCSP instance \((G, T, p, r)\) with \( G = (V, A) \) depicted in Figure B1.

A proof from [2] intends to show that \( v_{LP}(\text{RNCut}) \leq v_{LP}(\text{RSA}) \) holds. For this purpose, the authors consider an arbitrary solution \( x \in \Pi_{LP}(\text{RNCut}) \) 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

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

Figure B2 shows an auxiliary support graph of the instance illustrated by Figure B1. It is possible to send a flow with flow value \( \Tilde{x}(v) \) 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 \( \Tilde{x}(v) \) sent along arc \((j, l)\). Define the arc variables \( \Tilde{y}(j, l), (j, l) \in A \), of

\[ \text{FIGURE B1} \quad \text{Directed RMWCSP instance} \]

\[ \text{FIGURE B2} \quad \text{Illustration of an auxiliary support graph } G' \text{ corresponding to the instance in Figure B1 regarding the optimal solution } \Tilde{x}(v) = 0.5, v \in V \setminus T, \text{ and } \Tilde{x}(t) = 1, t \in T, \text{ to the RNCut formulation} \]
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 B2 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 [2] 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 [31]: 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.