The Maximum Weight Connected Subgraph Problem

Eduardo Álvarez-Miranda and Ivana Ljubić and Petra Mutzel

Abstract The *Maximum (Node-) Weight Connected Subgraph Problem* (MWCS) searches for a connected subgraph with maximum total weight in a node-weighted (di)graph. In this work we introduce a new integer linear programming formulation built on node variables only, which uses new constraints based on node-separators. We theoretically compare its strength to previously used MIP models in the literature and study the connected subgraph polytope associated with our new formulation. In our computational study we compare branch-and-cut implementations of the new model with two models recently proposed in the literature: one of them using the transformation into the Prize-Collecting Steiner Tree problem, and the other one working on the space of node variables only. The obtained results indicate that the new formulation outperforms the previous ones in terms of the running time and in terms of the stability with respect to variations of node weights.

1 Introduction

The *Maximum (Node-) Weight Connected Subgraph Problem* (MWCS) is the problem of finding a connected subgraph with maximum total weight in a node-weighted (di)graph. It belongs to the class of network design problems and has applications in various different areas such as forestry, wildlife preservation planning, systems biology, computer vision, and communication network design.

Eduardo Álvarez-Miranda
Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Italy, e.alvarez@unibo.it

Ivana Ljubić
Department of Statistics and Operations Research, University of Vienna, Austria, ivana.ljubic@univie.ac.at

Petra Mutzel
Department of Computer Science, TU Dortmund, Germany, petra.mutzel@tu-dortmund.de

Lee and Dooly [18] introduced a cardinality-constrained version of the problem for building a designed fiber-optic communication network over time, where the given node weights reflect their degree of importance. They defined the *maximum-weight connected graph problem* for an undirected graph with given node weights, in which they search the connected subgraph of maximum weight consisting of exactly a predescribed number of nodes. The same problem version was considered already in [14] (the authors called it *Connected k-Subgraph Problem*) for a Norwegian off-shore oil-drilling application.

Another application arises in the area of system biology [8, 22, 1]. Yamamoto et al. [22] suggest the cardinality-constrained MWCS in order to detect core source components in gene networks, which seem to be responsible for the difference between normal cells and mutant cells. The input graphs are constructed from gene regulation networks combined with gene expression data provided as node weights. Maximum weight connected subgraphs are considered to be good candidates for these core source components. A directed version of the MWCS has been considered in Backes et al. [1], where the most deregulated connected subnetwork in regulatory pathways with the highest sum of node scores (arising from expression data) is searched. In their model, they call a subgraph connected if all the nodes are reachable from one node, also called the *root* in the subgraph. The detected roots are likely to be the molecular *key-players* of the observed deregulation.

A budgeted version arises in conservation planning, where the task is to select land parcels for conservation to ensure species viability, also called *corridor design* (see, e.g. [7]). Here, the nodes of the graph do not only have node weights associated with the habitat suitability but also some costs, and the task is to design wildlife corridors that maximize the suitability with a given limited budget. Also in forest planning, the MWCS arises as a subproblem, e.g., for designing a contiguous site for a natural reserve or for preserving large contiguous patches of mature forest [3].

A surprising application of the MWCS arises in activity detection in video sequences. Here, a 3D graph is constructed from a video in which the nodes correspond to local video subregions and the edges to their proximity in time and space. The node weights correspond to the degree of activity of interest, and so the maximum weight connected subgraph corresponds to the portion of the video that maximizes a classifier's score [4].

All the above mentioned applications have in common that the MWCS arises with node weights only. In many papers, the MWCS has been solved by transforming the given instance to the *Prize-Collecting Steiner Tree Problem*. Here, the given graph has non-negative node weights and negative edge costs, and the task is to find a maximum weight subtree, where the weight is computed as the sum of the node and edge weights in the subtree. The Prize-Collecting Steiner Tree Problem has been studied intensively in the literature (see, e.g., [16, 20]), and the publicly available branch-and-cut (B&C) code of [20] is used in many recent applications to solve the underlying problems to optimality.

However, in their recent work, Backes et al. [1] attack the MWCS directly, which has the advantage to avoid variables for the arcs. The authors suggest a new integer linear programming formulation which is based on node variables only. The inten-

tion of our research was to study the MWCS straightly, and to suggest tight MIP formulations that improve the MIP models from the literature in theory and practice.

Our Contribution: We propose a new MIP model for the MWCS based on the concept of node separators in digraphs. We provide a theoretical and computational comparison of the new model with other models recently used in the literature. We show that the new model has the advantage of using only node variables while preserving the tight LP bounds of the Prize-Collecting Steiner Tree (PCStT) model. Furthermore, we study the connected subgraph polytope and show under which conditions the newly introduced inequalities are facet defining. In an extensive computational study, we compare different MIP models on a set of benchmark instances used in systems biology and on an additional set of network design instances. The obtained results indicate that the new formulation outperforms the previous ones in terms of the running time and in terms of the stability with respect to variations of node weights.

The paper is organized as follows. Section 2 contains a formal definition of the MWCS and some complexity results. The following Sections provide four different MIP formulations and polyhedral studies. Our B&C algorithm and the practical experiments are discussed in Section 5.

2 The Maximum Weight Connected Subgraph Problem

In this section we formally introduce the MWCS for directed graphs and discuss some complexity results.

Definition 1 *(The Maximum Weight Connected Subgraph Problem, MWCS) Given a digraph $G = (V,A)$, $|V| = n$, with node weights $p : V \to \mathbb{Q}$, the MWCS is the problem of finding a connected subgraph $T = (V_T, A_T)$ of G, that maximizes the score $p(T) = \sum_{v \in V_T} p_v$ and such that there exists a node $i \in V_T$ (called* root *or* key player *) such that every other node $j \in V_T$ can be reached from i by a directed path in T.*

The MWCS in undirected graphs is to find a connected subgraph T that maximizes the score $p(T)$. However, if $G = (V,E)$ is an undirected graph, without loss of generality we will consider its bidirected counterpart (V,A) where A is obtained by replacing each edge by two oppositely directed arcs. Hence, it is sufficient to present results that hold for digraphs (which are more general), and the corresponding results for undirected graphs can be easily derived from them. We assume that in our MWCS instances always positive and negative node weights are present, otherwise, the solution would be trivial. Observe that any feasible solution of the MWCS contains a tree with the same solution value. Hence it is equivalent to search a maximum node-weighted tree in the given graph.

Furthermore, it can be distinguished between the *rooted* and *unrooted* MWCS, i.e., a root node r can be pre-specified or not. In this paper we will concentrate on the unrooted MWCS, or simply the MWCS in the rest of the paper.

Regarding the complexity of the MWCS, it has been shown that the problem is NP-hard (in the supplementary documentation of the paper by [15], the authors provide an NP-hardness proof sketched by R. Karp). Since it is possible to translate the problem to the Prize-Collecting Steiner tree problem, all its polynomially solvable cases carry over to the MWCS. E.g., the PCStT is solvable in polynomial time for the graph class of bounded treewidth [2].

Furthermore, one can show that the following result holds even when the MWCS is defined on undirected graphs:

Proposition 1 *It is NP-hard to approximate the optimum of the MWCS within any constant factor* $0 < \varepsilon < 1$.

Proof. For a given MWCS instance, let *APP* be the objective function value of an approximate solution, and let *OPT* be the optimal solution value. Recall that for a given constant $0 < \varepsilon < 1$, a given problem can be approximated within factor ε if and only if $APP/OPT \geq \varepsilon$, for any problem instance. To prove this result for the MWCS it is sufficient to make a reduction from the SAT problem that works similarly to the one given in [9, cf. Theorem 4.1]. By doing so, we can show that for a given formula ϕ for SAT, we can build an instance $G = (V,E)$ of the MWCS in polytime, such that: (i) if ϕ is a yes-instance, then the optimal MWCS solution on G has value $\varepsilon(1+\varepsilon^3)$, and (ii) if ϕ is a no-instance, then the optimal MWCS solution on G has value ε^2. □

Some applications consider the *cardinality-constrained MWCS*, where the task is to find a connected subgraph with K nodes. Hochbaum and Pathria [14] have shown that this problem version is NP-hard even if all node weights are 0 or 1 and the graph is either bipartite or planar. For trees and for complete layered DAGs, it is solvable in polynomial time via dynamic programming [14, 19]. Observe that for this problem version, the node weights can be assumed to be all positive, and the maximization variant and the minimization variant are equivalent. Goldschmidt [13] noted that no approximation algorithm is known with a factor better than $O(K)$, and such an algorithm is almost trivial to find. The cardinality-constrained MWCS (and also the MWCS) can be solved by translating it into the edge-weighted version, which has been studied as the *k-Minimum Spanning Tree Problem* (*k*-MST) or *k-Cardinality Tree Problem* in the literature (see, e.g., [10, 6]).

3 MIP Formulations for the MWCS

In this section we revise three MIP models for the MWCS recently presented in the literature, and propose a novel approach based on the concept of node separators in digraphs.

The MIP formulations considered in this paper are based on the observation that if there is a path between i and any other node in $T = (V_T, A_T)$, then we will search for a subgraph which is an arborescence rooted at $i \in V_T$. In our models, two types of binary variables will be used to describe a feasible MWCS solution $T = (V_T, A_T)$: binary variables y_i associated to nodes $i \in V$ will be set to one iff $i \in V_T$, and additional binary variables x_i will be set to one iff the node $i \in V$ is the key player, i.e., if it is used as the root of the arborescence.

Notation and Preliminaries: A set of vertices $S \subset V$ ($S \neq \emptyset$) and its complement $\overline{S} = V \setminus R$ induce two directed cuts: $(S, \overline{S}) = \delta^+(S) = \left\{(i,j) \in A \mid i \in S, j \in \overline{S}\right\}$ and $(\overline{S}, S) = \delta^-(S) = \left\{(i,j) \in A \mid i \in \overline{S}, j \in S\right\}$. When there is an ambiguity regarding the graph in which the directed cut is considered, we will sometimes write δ_G instead of only δ to specify that the cut is considered w.r.t. graph G. For a set $C \subset V$, let $D^-(C)$ denote the set of nodes outside of C that have ingoing arcs into C, i.e., $D^-(C) = \{i \in V \setminus C \mid \exists (i,v) \in A, v \in C\}$.

A digraph G is called strongly connected (or simply, *strong*) if for any two distinct nodes k and ℓ from V, there exists a (k,ℓ) path in G. A node i is a cut point in a strong digraph G if there exists a pair of distinct nodes k and ℓ from V such that there is no (k,ℓ) path in $G - i$.

For two distinct nodes k and ℓ from V, a subset of nodes $N \subseteq V \setminus \{k,\ell\}$ is called (k,ℓ) *node separator* if and only if after eliminating N from V there is no (k,ℓ) path in G. A separator N is *minimal* if $N \setminus \{i\}$ is not a (k,ℓ) separator, for any $i \in N$. Let $\mathscr{N}(k,\ell)$ denote the family of all (k,ℓ) separators. Obviously, if $\exists (k,\ell) \in A$ or if ℓ is not reachable from k, we have $\mathscr{N}(k,\ell) = \emptyset$. Let $\mathscr{N}_\ell = \cup_{k \neq \ell} \mathscr{N}(k,\ell)$ be the family of all node separators with respect to $\ell \in V$ that we will refer to as ℓ-separators.

For binary variables $\mathbf{a} \in \{0,1\}^{|F|}$, we denote by $a(F')$ the sum $\sum_{i \in F'} a_i$ for any subset $F' \subseteq F$.

3.1 The Prize-Collecting Steiner Tree Model

In [8] the authors observed that the MWCS on undirected graphs is equivalent to the Prize-Collecting Steiner Tree Problem (PCStT), in the sense that there exists a transformation from the MWCS into the PCStT such that each optimal solution of the PCStT on the transformed graph corresponds to an optimal MWCS solution from the original graph. Recall that, given an undirected graph $H = (V_H, E_H)$ with non-negative node weights $\tilde{p}_v$ and non-negative edge costs $\tilde{c}_e$, the PCStT is the problem of finding a subtree T_H of H that maximizes the function $\sum_{v \in T_H} \tilde{p}_v - \sum_{e \in T_H} \tilde{c}_e$, i.e., the difference between the collected node prizes and edge costs. The transformation from the MWCS into the PCStT is given as follows: Given an input graph G of the MWCS we set $H := G$ and $w = \min_{v \in V} p_v$ (note, that $w < 0$). In order to get non-negative node weights, we set $\tilde{p}_v := p_v - w \; \forall v \in V$ and $\tilde{c}_e = -w$, for all $e \in E$. This transformation also works for digraphs, i.e., if H is a digraph, the PCStT consists of finding a subarborescence of H (rooted at some node $i \in V$) that maximizes the given

objective function. The transformation is correct, since any feasible solution is an arborescence, which has indegree 1 for every node, and the weight transformations neutralize each other.

We now present the MIP model proposed in [20] for the PCStT that is used for solving the MWCS after transforming it into the PCStT (see [8]). Consider a transformation from a (directed or undirected) PCStT instance into a rooted digraph $G_d = (V_d, A_d)$ that works as follows: If the input graph $G = (V, E)$ is undirected, then we create the arc set A by bidirecting each edge. In any case we now have a directed graph $G = (V, A)$. The vertex set $V_d = V \cup \{r\}$ contains the nodes of the input graph G *and an artificial root* vertex r. We add new arcs from the root r to nodes v whose out-degree is non-empty in order to get the arc set A_d i.e., $A_d = A \cup \{(r,v) \mid v \in V \text{ and } \delta^+(v) \neq \emptyset\}$. All arc weights are set to the weights of their undirected counterparts, and the weight of an arc $(r,v) \in A_d$ is set to w.

In the graph G_d, a subgraph $T_d = \left(V_{T_d}, A_{T_d}\right)$ that forms a directed tree rooted at r is called a *rooted Steiner arborescence*. It is a feasible solution of the PCStT if the out-degree of the root is equal to one. To model feasible Steiner arborescences in G_d, we will use two types of binary variables: (a) binary variables y_i introduced above associated to all nodes $i \in V$, and (b) binary variables z_{ij}, such that $z_{ij} = 1$ if arc (i, j) belongs to a feasible Steiner arborescence T_d and $z_{ij} = 0$ otherwise, for all $(i, j) \in A_d$.

The set of constraints that characterizes the set of feasible solutions of the unrooted PCStT is given by:

$$z(\delta^-(i)) = y_i, \quad \forall i \in V \setminus \{r\} \tag{1}$$

$$z(\delta^-(S)) \geq y_k, \quad \forall S \subseteq V \setminus \{r\},\ k \in S \tag{2}$$

$$z(\delta^+(r)) = 1. \tag{3}$$

The *in-degree* constraints (1) guarantee that the in-degree of each vertex of the tree is equal to one. The directed cut constraints (2) ensure that there is a directed path from the root r to each costumer k such that $y_k = 1$. The equality (3) makes sure that the artificial root is connected to exactly one of the nodes. Thus, the MWCS can be formulated using the following model that we will denote by $(PCStT)$:

$$\max \left\{ \sum_{v \in V} (p_v - w) y_v + \sum_{(i,j) \in A_d} w z_{ij} \mid (\mathbf{y}, \mathbf{z}) \text{ satisfies (1)-(3)},\ (\mathbf{y}, \mathbf{z}) \in \{0,1\}^{n+|A_d|} \right\}.$$

The $(PCStT)$ model uses node and arc variables (**y** and **z**) given that it relies on an equivalence with the PCStT. However, considering Definition 1 it seems more natural to find a formulation based only in the space of **y** variables since no arc costs are involved. In the next section we will discuss several models that enable elimination of arc variables in the MIP models.

3.2 Model of Backes et al. 2011

Recently, in [1] a new MIP model for the MWCS is introduced which avoids the explicit use of arc variables. Let $\mathscr{C}$ denote the family of all directed cycles in G. The new model, that we will denote by $(CYCLE)$, reads as follows:

$$x(V) = 1 \tag{4}$$
$$x_i \leq y_i, \quad \forall i \in V \tag{5}$$
$$y(D^-(i)) \geq y_i - x_i, \quad \forall i \in V \tag{6}$$
$$y(C) - x(C) - y(D^-(C)) \leq |C| - 1, \quad \forall C \in \mathscr{C} \tag{7}$$
$$(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{2n}. \tag{8}$$

Inequalities (4) make sure that one node is selected as a root, and inequalities (5) state that if the node is chosen as a root, it has to belong to the solution. Constraints (6) are the *in-degree constraints* – they ensure that for each node which is not the root, at least one of the incoming neighbors needs to be taken into the solution. In a directed acyclic graph, in-degree constraints are sufficient to guarantee connectivity, but in general, imposing only the in-degree constraints may allow solutions that consist of several disconnected components. To avoid this, cycle constraints (7) are added to guarantee connectivity. These constraints make sure that whenever all nodes from a cycle are taken in a solution, and none of them is set as the root, at least one of the neighboring nodes from $D^-(C)$ has to be taken as well.

Observation 1 *Constraints* (7) *are redundant for those* $C \in \mathscr{C}$ *such that* $C \cup D^-(C) = V$.

To see this, observe that using the root constraint (4), the cycle constraints (7) can be rewritten as follows:

$$y(C) \leq y(D^-(C)) + |C| - 1 + x(C) = y(D^-(C)) + |C| - x(D^-(C)),$$

which is always satisfied by the model due to constraints (5) and $y_i \leq 1$, for all $i \in V$.

In this model an artificial root node r is not explicitly introduced. However, it is not difficult to see that for any feasible MWCS solution there is a one-to-one mapping between variables z_{ri} introduced above and the variables x_i, for all $i \in V$.

The following result shows that the $(CYCLE)$ model provides very weak upper bounds, in general.

Lemma 1. *Given an instance of the MWCS, let OPT be the value of the optimal solution, and let UB be the upper bound obtained by solving the LP relaxation of the* $(CYCLE)$ *model. Then, there exist MWCS instances for which* $UB/OPT \in O(n)$.

Proof. Consider an example given in Fig. 1. The variables of the LP relaxation of the $(CYCLE)$ model are set as follows: $y_i = x_i = 0$ for the nodes i with negative weights; $y_i = 1/2$ and $x_i = 0$ for the nodes i in the 2-cycles, and $x_i = y_i = 1$ for the node in the center. There are $K_n = (n-1)/3 \in O(n)$ branches in this graph. We have $UB = K_n M + 2M$ and $OPT = 2M$, which concludes the proof. □

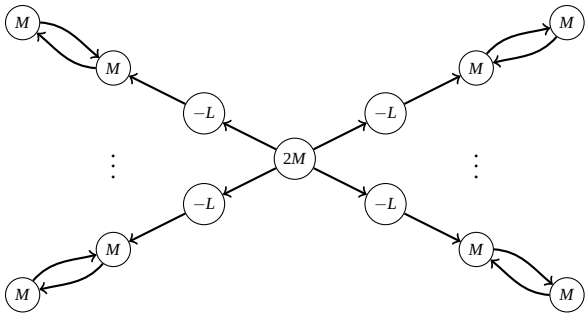


Fig. 1 An example showing that the LP bounds of the $(CYCLE)$ model can be as bad as $O(n)$. The labels of nodes represent their weights: $M > 0$ and $L >> M$.

3.3 A Model Based on (k,ℓ) Node Separators

We now present an alternative approach to model the MWCS in the space of $(\mathbf{x},\mathbf{y})$ variables that relies on the constraints that have been recently used by [11] and [3] to model connectivity in the context of sheet metal design and forest planning, resp. Notice that for an arbitrary pair of distinct nodes (k,ℓ) in G, if ℓ is taken into the solution and k is chosen as root, then either (i) there is a direct arc from k to ℓ, or (ii) at least one node from any (k,ℓ) separator $N \in \mathcal{N}(k,\ell)$ has to be taken into the solution. The latter fact can be stated using the following inequalities that we will refer to as *node-separator constraints*:

$$y(N) - x(N) \geq y_\ell + x_k - 1, \quad \forall k,\ell \in V,\ \ell \neq k,\ N \in \mathcal{N}(k,\ell). \tag{9}$$

If the nodes k and ℓ are connected by an arc, then $\mathcal{N}(k,\ell) = \emptyset$, in which case we need to consider the in-degree inequalities (6) to make sure k is connected to ℓ. Thus, we can formulate the unrooted MWCS as

$$(CUT)_{k,\ell} \qquad \max \left\{ \sum_{v \in V} p_v y_v \mid (\mathbf{x},\mathbf{y}) \text{ satisfies (4)-(6), (9) and } (\mathbf{x},\mathbf{y}) \in \{0,1\}^{2n} \right\}.$$

Inequalities (9) can be separated in polynomial time in a support graph that splits nodes into arcs. Given a fractional solution $(\tilde{x},\tilde{y})$, for each pair of nodes (k,ℓ) such that $\tilde{y}_\ell + \tilde{x}_k - 1 > 0$ we generate a graph $G_{k\ell}$ in which all nodes $i \neq k,\ell$ are replaced by arcs. Arc capacities are then set to 1, except for the arcs associated to nodes, whose capacities are set to $\tilde{y}_i - \tilde{x}_i$. If the maximum flow that can be sent from k to ℓ in $G_{k\ell}$ is less than $\tilde{y}_\ell + \tilde{x}_k - 1 > 0$, we have detected a violated inequality of type (9).

Using the root constraint (4), inequalities (9) can also be reformulated as follows:

$$y(N) \geq y_\ell + x(N \cup \{k\}) - 1 \ \Rightarrow y(N) + x(V \setminus (N \cup \{k,l\})) \geq y_\ell - x_\ell,$$

which can be interpreted as follows: If node ℓ is in the solution and it is not the root, then for each $k \in V$ such that $\mathcal{N}(k,\ell) \neq \emptyset$ and each $N \in \mathcal{N}(k,\ell)$, either one of the nodes from N is part of the solution, or none of the nodes from $N \cup \{k\}$ is chosen as the root node.

Inequalities (9) are quite intuitive, however they are not facet defining. In the next section we will show how the (k,ℓ) node separator constraints can be lifted to obtain facet defining inequalities.

3.4 A Model Based on Generalized Node Separator Inequalities

Observe that the inequality (9) can be lifted as follows: Assume that $N \in \mathcal{N}(k,\ell)$ also separates another node $k' \neq k$ from ℓ. Since at most one node can be set as a root, the right-hand side of (9) can be increased as follows: $y(N) - x(N) \geq y_\ell + x_k + x_{k'} - 1$. In fact, this motivates us to introduce a generalized family of node separator inequalities, that can be obtained by a parallel lifting of (9).

Generalized Node-Separator Inequalities: Let ℓ be an arbitrary node in V and let $N \in \mathcal{N}_\ell$ be an arbitrary ℓ-separator. Let $W_{N,\ell}$ be the set of nodes i such that there is a directed (i,ℓ)-path in $G - N$. More formally:

$$W_{N,\ell} = \{i \in V \setminus N \mid \exists (i,\ell) \text{ path } P \text{ in } G - N\} \cup \{\ell\}.$$

Then, for any feasible MWCS solution, the following has to be satisfied: if node ℓ is part of a solution, then either the root of the solution is in $W_{N,\ell}$, or, otherwise, at least one of the nodes from N has to be taken. Hence, the following inequalities, that we will refer to as *generalized node-separator inequalities*, are valid for the MWCS:

$$y(N) + x(W_{N,\ell}) \geq y_\ell, \quad \forall \ell \in V,\ N \in \mathcal{N}_\ell \tag{gNSep}$$

Notice that the in-degree inequalities (6) are a subfamily of (gNSep): The in-degree inequality can be rewritten as $\sum_{j \in D^-(\ell)} y_j + x_\ell \geq y_\ell$, i.e., they are a special case of the generalized node-separator cuts for $N = D^-(\ell)$ in which case $W_{N,\ell} = \{\ell\}$. In order to see that (gNSep) are lifted inequalities (9), notice that (gNSep) can be rewritten as follows:

$$y(N) - x(N) \geq y_\ell + x(V \setminus (N \cup W_{N,\ell})) - 1, \quad \forall \ell \in V,\ N \in \mathcal{N}_\ell.$$

Together with this observation this proves that the following model is a valid MIP formulation for the MWCS:

$$(CUT) \qquad \max \left\{ \sum_{v \in V} p_v y_v \mid (\mathbf{x},\mathbf{y}) \text{ satisfies (4)-(5), (gNSep) and } (\mathbf{x},\mathbf{y}) \in \{0,1\}^{2n} \right\}.$$

Proposition 2 *Generalized node-separator inequalities can be separated in polynomial time.*

Proof. Consider an auxiliary support graph in which the nodes are splitted as follows: each node $i \in V$ is replaced by an arc (i_1, i_2). All ingoing arcs into i are now connected to i_1, all outgoing arcs from node i are now connected to i_2. In other words, we create a graph $G' = (V', A')$ such that $V' = \{i_1 \mid i \in V\} \cup \{i_2 \mid i \in V\} \cup \{r\}$ (r is an artificial root), $A' = \{(i_2, j_1) \mid (i,j) \in A\} \cup \{(i_1, i_2) \mid i \in V\} \cup \{(r, i_1) \mid i \in V\}$. For a given fractional solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ arc capacities in G' are defined as:

$$cap_{uv} = \begin{cases} \tilde{y}_i, & \text{if } u = i_1, v = i_2, i \in V, \\ \tilde{x}_i, & \text{if } u = r, v = i_1, i \in V, \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

We calculate the maximum flow on G' between r and (ℓ_1, ℓ_2) in G' for a node ℓ such that $\tilde{y}_\ell > 0$. To check whether there are violated inequalities of type (gNSep), it only remains to show that (i) every minimum cut $(\overline{S}, S)$ in G' such that the corresponding flow is less than $\tilde{y}_\ell$ corresponds to a (gNSep) inequality for the given $\ell \in V$ and some $N \in \mathcal{N}_\ell$, or (ii) that a corresponding violated (gNSep) cut can be generated from $(\overline{S}, S)$ in polynomial time. Observe that any minimum cut $(\overline{S}, S)$ in G' which is smaller than $\tilde{y}_\ell$ can be represented as union of arcs adjacent to the root, plus union of arcs of type (i_1, i_2). Hence, each $(\overline{S}, S)$ cut implies the following inequalities:

$$\sum_{(r,j) \in \delta^-(S)} x_j + \sum_{(i_1, i_2) \in \delta^-(S)} y_i \geq y_\ell. \tag{11}$$

We can now define a partitioning (U, N, W) of the node set V such that:

$$W = \{i \in V \mid i_1, i_2 \in S\}, \quad N = \{i \in V \mid i_1 \notin S, i_2 \in S\}, \quad U = V \setminus (W \cup N).$$

Rewriting the inequality (11), we obtain: $x(W) + y(N) \geq y_\ell$. Observe that $U \neq \emptyset$. Indeed, if $U = \emptyset$ then $N \cup W = V$, but then we have $x(N) + y(W) \geq x(V) = 1 \geq \tilde{y}_\ell$, i.e., such cuts will never be violated. Hence, given the proper partition (U, N, W), the set N is obviously a (k, ℓ) separator for any $k \in U$ (after removing (r, i_1) arcs from G', the arcs $(i_1, i_2) \in \delta^-(S)$ are arc-separators that separate U from the rest of the graph). If W contains only nodes that can reach ℓ in $G - N$, then inequality (11) belongs to the (gNSep) family. Otherwise we reverse all arcs in $G - N$ and perform a breadth-first search from ℓ. All nodes that can be reached from ℓ (notice that they cannot belong to U), by definition, determine the set $W_{N,\ell}$. If the original cut (11) was violated, the new one with the left-hand side equal to $y(N) + x(W_{N,\ell})$ will be violated as well. □

3.5 Some More Useful Constraints

In this section we present additional constraints that are useful for practically solving MWCS instances.

Connected Component Inequalities: In some applications of the MWCS, a K-cardinality constraint is imposed: $\sum_{i \in V} y_i = K$. For a given node $k \in V$, let P_k contain all the nodes that are further than $K-1$ hops away from k. In that case, the following inequalities are valid for the MWCS:

$$x_k + y_\ell \leq 1, \quad \forall \ell \in P_k. \tag{12}$$

Rewriting the connected component cuts, we obtain:

$$\sum_{j \neq k} x_j \geq y_\ell, \quad \forall \ell \in P_k,$$

these constraints can be further strengthened by down lifting the coefficients of the left-hand side. Whenever node ℓ is in the solution, then either ℓ is the root, or the root cannot be more than $K-1$ hops away from ℓ. Let W_ℓ be the set of such potential root nodes including ℓ. We have

$$x(W_\ell) \geq y_\ell, \quad \forall \ell \in V.$$

Out-Degree Inequalities: The following set of inequalities state that whenever a node i such that $p_i \leq 0$ is taken into a solution, this is because it leads us to another node with positive weights:

$$y(D^+(i)) \geq y_i, \quad \forall i \in V \text{ s.t. } p_i \leq 0. \tag{13}$$

Observe that these constraints are not valid if K-cardinality constraints are imposed.

Symmetry-Breaking Inequalities: In case the input graph is undirected, there exist many equivalent optimal solutions with different orientations. In order to break those symmetries, we can impose the following constraint that chooses the node with the smallest index to be the root of the subgraph:

$$x_j + y_i \leq 1, \quad \forall i < j. \tag{14}$$

4 Polyhedral Study

Let $\mathscr{P}$ denote the connected subgraph (CS) polytope in the space of $(\mathbf{x}, \mathbf{y})$ variables:

$$\mathscr{P} = \text{conv}\{(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{2n} \mid (\mathbf{x}, \mathbf{y}) \text{ satisfies (4), (5), (gNSep)}\}.$$

In this section we compare the proposed MIP formulations with respect to their quality of LP bounds and we show that, under certain conditions, the newly introduced generalized node-separator inequalities are facet defining for the CS polytope.

4.1 Theoretical Comparison of MIP Models

Let $\mathscr{P}_{\text{LP}}(.)$ denote the polytope of the LP relaxations of the MIP models presented above obtained by replacing integrality conditions by $0 \leq x_i, y_i \leq 1$, for all $i \in V$, and let $v_{LP}(.)$ be the optimal LP values of the associated MIP relaxations. For the $\mathscr{P}_{\text{LP}}(PCStT)$ polytope, we set $Proj_{(x,y)}(\mathscr{P}_{\text{LP}}(PCStT)) = \{(\mathbf{x},\mathbf{y}) \in \{0,1\}^{2n} \mid x_i = z_{ri}$ and $(y,z) \in \mathscr{P}_{\text{LP}}(PCStT)\}$. We can show that:

Proposition 3 *We have:*

1. $Proj_{(x,y)}(\mathscr{P}_{\text{LP}}(PCStT)) = \mathscr{P}_{\text{LP}}(CUT) \subsetneq \mathscr{P}_{\text{LP}}(CUT_{k\ell})$ *and* $\mathscr{P}_{\text{LP}}(CUT) \subsetneq \mathscr{P}_{\text{LP}}(CYCLE)$.
2. *Moreover, there exist MWCS instances such that* $v_{LP}(CYCLE)/v_{LP}(CUT) \in O(n)$.
3. *The polytopes* $\mathscr{P}_{\text{LP}}(CYCLE)$ *and* $\mathscr{P}_{\text{LP}}(CUT_{k\ell})$ *are not comparable.*

Proof. 1. $Proj_{(x,y)}(\mathscr{P}_{\text{LP}}(PCStT)) = \mathscr{P}_{\text{LP}}(CUT)$: We first show that $Proj_{(x,y)}(\mathscr{P}_{\text{LP}}(PCStT)) \subseteq \mathscr{P}_{\text{LP}}(CUT)$. Let $(\hat{\mathbf{y}},\hat{\mathbf{z}})$ be a feasible solution for the relaxation of the *PCStT* model, we will show that the solution $(\hat{\mathbf{x}},\hat{\mathbf{y}})$ such that $\hat{x}_i = \hat{z}_{ri}$ belongs to $\mathscr{P}_{\text{LP}}(CUT)$. Let $\ell \in V$ be an arbitrary node such that $\hat{y}_\ell > 0$, choose some $N \in \mathscr{N}_\ell$ and consider the associated $W_{N,\ell} \subset V$. Let G_d be the corresponding directed instance of the PCStT with the root r (cf. Section 3.1). Consider now a cut $(\overline{W}_d, W_d)$ in G_d where $W_d = N \cup W_{N,\ell}$. We have: $\delta^-_{G_d}(W_d) = \{(r,i) \in A_d \mid i \in W_{N,\ell}\} \cup Rest$, where $Rest = \{(j,i) \in A_d \mid j \in \overline{W}_d, i \in N\}$. Observe that $Rest \subseteq \delta^-_{G_d}(N) \subseteq \cup_{i\in N}\delta^-_{G_d}(i)$. Therefore, we have:

$$\hat{\mathbf{y}}(N) = \sum_{i\in N} \hat{\mathbf{z}}(\delta^-_{G_d}(i)) \geq \hat{\mathbf{z}}(\delta^-_{G_d}(N)) \geq \hat{\mathbf{z}}(Rest). \tag{15}$$

Since $(\overline{W}_d, W_d)$ is a Steiner cut in G_d, it holds that $\hat{\mathbf{z}}(\delta^-_{G_d}(W_d)) \geq \hat{y}_\ell$. This, together with (15) implies:

$$\hat{\mathbf{y}}(N) + \hat{\mathbf{x}}(W_{N,\ell}) \geq \hat{\mathbf{z}}(Rest) + \hat{\mathbf{x}}(W_{N,\ell}) = \hat{\mathbf{z}}(\delta^-_{G_d}(W_d)) \geq \hat{y}_\ell.$$

To show that $\mathscr{P}_{\text{LP}}(CUT) \subseteq Proj_y(\mathscr{P}_{\text{LP}}(PCStT))$ consider an LP solution $(\check{\mathbf{y}},\check{\mathbf{x}}) \in \mathscr{P}_{\text{LP}}(CUT)$. We will construct a solution $(\hat{\mathbf{y}},\hat{\mathbf{z}}) \in \mathscr{P}_{\text{LP}}(PCStT)$ such that $\check{\mathbf{y}} = \hat{\mathbf{y}}$ and $\hat{z}_{rj} = \check{x}_j$, $\forall j \in V$. On the graph G' (see Proof of Proposition 2) with arc capacities of (i_1,i_2) set to $\check{y}_i$ for each $i \in V$, arc capacities of (r,j_1) set to $\check{x}_j$, and capacities set to 1 for the remaining arcs, we are able to send $\check{y}_\ell$ units of flow from the root r to every $\ell_1 \in V'$ such that $\check{y}_\ell > 0$. Let f^k_{ij} denote the amount of flow of commodity k, associated with $k_1 \in V'$, sent along an arc $(i,j) \in A'$. Let $\mathbf{f}$ be the minimal feasible

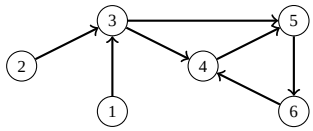


Fig. 2 An example showing that $\mathscr{P}_{\mathrm{LP}}(CUT_{k\ell}) \not\subseteq \mathscr{P}_{\mathrm{LP}}(CYCLE)$. The LP solution $y_4 = y_5 = y_6 = 1$, $y_1 = y_2 = y_3 = x_1 = x_2 = 1/2$ is feasible for the $(CUT_{k\ell})$ model and infeasible for $(CYCLE)$.

multi-commodity flow on G' (i.e., the effective capacities on G' used to route the flow cannot be reduced without violating the feasibility of this flow). We now define the values of $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ as follows: $\hat{z}_{rj} = \check{x}_j \,, \forall j \in V$ and

$$\hat{z}_{ij} = \begin{cases} \max_{k \in V} f^k_{i_2 j_1}, & i, j \in V \\ \max_{k \in V} f^k_{r j_1}, & i = r, j \in V \end{cases} \,, \forall (i,j) \in A; \;\; \hat{y}_i = \hat{\mathbf{z}}(\delta^-(i)) \,, \forall i \in V.$$

Obviously, the constructed solution $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ is feasible for the $(PCStT)$ model and, due to the assumption that $\mathbf{f}$ is minimal feasible, it follows that $\check{\mathbf{y}} = \hat{\mathbf{y}}$ and $\check{\mathbf{x}}$ is equivalent to $\hat{\mathbf{z}}$, which concludes the proof.

$\mathscr{P}_{\mathrm{LP}}(CUT) \subsetneq \mathscr{P}_{\mathrm{LP}}(CYCLE)$: Let $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ be an arbitrary point from $\mathscr{P}_{\mathrm{LP}}(CUT)$. In order to prove that $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \in \mathscr{P}_{\mathrm{LP}}(CYCLE)$ we only need to show that constraints (7) are satisfied (recall that in-degree inequalities (6) are contained in (gNSep)). Given the Observation 1, it is sufficient to consider cycles C such that $C \cup D^-(C) \subset V$. Since for any such cycle C the set $D^-(C)$ defines a separator for any node $\ell \in C$, from constraints (gNSep) we have that $\hat{y}(D^-(C)) + \hat{x}(C) \geq \hat{y}_\ell$. For the remaining nodes $j \in C$, $j \neq k$, we apply the bounds $1 \geq \hat{y}_j$. Summing up together these $|C|$ inequalities, we obtain (7).

2. Consider the example given in Fig. 1 for which the (CUT) model finds the optimal solution.

3. The example given in Fig. 1 shows an instance for which the LP solution is feasible for the $(CYCLE)$ and infeasible for the $(CUT_{k\ell})$ model. The example given in Fig. 2 shows an instance for which the LP solution is feasible for the $(CUT_{k\ell})$ and infeasible for the $(CYCLE)$ model. □

4.2 Facets of the CS Polytope

In this section we establish under which conditions some of the presented inequalities are facet defining for the CS polytope.

Lemma 2. *If G is a strong digraph, then the dimension of the polytope $\mathscr{P}$ is* $\dim(\mathscr{P}) = 2n - 1$.

Proof. We will construct the set of $2n$ feasible, affinely independent solutions as follows: Since G is strong, we can find n spanning arborescences by choosing each

$i \in V$ as a root. That way, we build n affinely independent solutions. In addition, consider n single node solutions (for each $i \in V$), in which we have $x_i = y_i = 1$ and all remaining $x_j = y_j = 0$, for all $j \neq i$. The matrix obtained by merging the characteristic vectors of these solutions has full rank, $2n$. □

Lemma 3. *Trivial inequalities $x_i \geq 0$ are facet defining if G is strong and i is not a cut point in G.*

Proof. Consider a family $\mathscr{T}$ of spanning arborescences on the set $V \setminus \{i\}$ in which each $j \neq i$ is taken once as a root. This is possible because $G - i$ remains a strong digraph. There are $n-1$ such solutions, and they are affinely independent. Add now to $\mathscr{T}$ single node solutions, for each $j \in V \setminus \{i\}$. Finally, add to $\mathscr{T}$ a spanning arborescence in G with a root $j \neq i$. The matrix associated to incidence vectors from $\mathscr{T}$ has full rank, $2n-1$. □

Lemma 4. *Trivial inequalities $y_i \leq 1$ are facet defining if G is strong.*

Proof. Consider a spanning arborescence T rooted at i. We will then apply a *pruning technique* in order to generate n affine independent feasible MWCS solutions. We start with T in which case $\mathbf{y}$ consists of all ones. We iteratively remove one by one leaves from T, until we end up with a single root node i. Thereby, we generate a family $\mathscr{T}$ of n affinely independent solutions. We then add to $\mathscr{T}$ $n-1$ solutions obtained by choosing a spanning arborescence rooted at j, for all $j \neq i$. The matrix associated to incidence vectors from $\mathscr{T}$, has full rank, $2n-1$. □

Notice that $y_i \geq 0$ are not facet defining inequalities because $y_i = 0$ implies $x_i = 0$. Similarly, $x_i \leq 1$ do not define facets of $\mathscr{P}$ because they are dominated by $x_i \leq y_i$.

Lemma 5. *Coupling inequalities $y_i \geq x_i$ are facet defining if G is strong and i is not a cut point in G.*

Proof. Construct a family $\mathscr{T}$ of n affinely independent solutions by applying pruning to a spanning arborescence rooted at i. Add then to $\mathscr{T}$ additional $n-1$ arborescences on the set $V \setminus \{i\}$ in which each $j \neq i$ is taken once as a root (this is possible because $G - i$ remains strong). The matrix associated to incidence vectors from $\mathscr{T}$, has full rank, $2n-1$. □

Proposition 4 *Given $\ell \in V$ and $N \in \mathscr{N}_\ell$, the associated* (gNSep) *inequality is facet defining if G is strong, N is a minimal ℓ-node separator and the subgraph induced by $W_{N,\ell}$ ($|W_{N,\ell}| \geq 2$) is strong.*

Proof. We prove the result by the indirect method. Let $F(\ell, N) = \{(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{2n} \mid y(N) + x(W_{N,\ell}) = y_\ell\}$. Consider a facet defining inequality of the form $\mathbf{ax} + \mathbf{by} \geq a_0$. We will show that if all points in $F(\ell, N)$ satisfy

$$\mathbf{ax} + \mathbf{by} = a_0, \tag{16}$$

then (16) is a positive multiple of (gNSep). Consider $\ell' \in W$, $\ell' \neq \ell$. A path from ℓ to ℓ', completely contained in $W_{N,\ell}$ and rooted at ℓ exists in G ($W_{N,\ell}$ is strong) and it is

a feasible MWCS solution that belongs to $F(\ell,N)$. Let $(\mathbf{x}^1,\mathbf{y}^1)$ be the characteristic vector of this path. A subpath obtained after removing ℓ' from this path, also rooted at ℓ, is another feasible solution from $F(\ell,N)$, and let $(\mathbf{x}^2,\mathbf{y}^2)$ be the corresponding characteristic vector. We have: $\mathbf{a}\mathbf{x}^1+\mathbf{b}\mathbf{y}^1-\mathbf{a}\mathbf{x}^2-\mathbf{b}\mathbf{y}^2=0$. Therefore we have $b'_\ell=0$, for all $\ell'\in W$, $\ell'\neq\ell$. Consider now a node $k\in U=V\setminus(N\cup W_{N,\ell})$. To show that $b_k=0$, for all $k\in U$, we distinguish the following cases:
(1) If $D^-(k)\cap U\neq\emptyset$, then there exists an arc (k',k), $k'\in U$ that builds a feasible MWCS solution B from $F(\ell,N)$. Also, the single node solution $B'=\{k'\}$ belongs to $F(\ell,N)$. After subtracting the equations (16) with the substituted characteristic vectors of B and B', we obtain $b_k=0$.
(2) If there exists an arc $(i,k)\in A$ for some $i\in N$, then, consider a path P from i to ℓ that does not cross $N\cup U$ (such P exists because N is minimal) and a path $P'=P\cup\{(i,k)\}$, in both of them we set i as root. Both P and P' belong to $F(\ell,N)$. After subtracting the equations (16) with the substituted characteristic vectors of P and P', we obtain $b_k=0$.
(3) Finally, if there exists an arc $(j,k)\in A$ for some $j\in W_{N,\ell}$, we consider a path Q from ℓ to j in $W_{N,\ell}$ (such path exists because $W_{N,\ell}$ is strong) and a path $Q'=Q\cup\{(j,k)\}$. Both Q and Q' belong to $F(\ell,N)$. After subtracting the equation (16) with the substituted characteristic vectors of Q and Q', we obtain $b_k=0$. Hence, the equation (16) can be rewritten as $\mathbf{a}\mathbf{x}+\sum_{i\in N\cup\{\ell\}}b_ix_i=a_0$. Notice that a single node solution $\{k\}$ belongs to $F(\ell,N)$, for each $k\in U$. By plugging the associated vector into (16), it follows that $a_k=a_0$, for all $k\in U$. Consider now two spanning arborescences in $W_{N,\ell}$, one rooted at ℓ, the other rooted at arbitrary $\ell'\neq\ell$ (this is possible, because $W_{N,\ell}$ is strong). After subtracting the equation (16) with the substituted characteristic vectors of those two arborescences, we obtain $a_{\ell'}=a_\ell=\alpha$, for all $\ell'\in W_{N,\ell}$. Since $N\in\mathcal{N}_\ell$ and it is minimal, for each $i\in N$ there exist $k\in U$ such that there exist a path P_k from k to ℓ that crosses N exactly at the node i. Let P'_k be a subpath of P_k from i to ℓ. Both paths belong to $F(\ell,N)$ and after subtracting the associated equations (16), it follows that $a_i=a_k$, and hence $a_i=a_0$, for all $i\in N$.

So far, (16) can be rewritten as $a_0x(\overline{W}_{N,\ell})+\alpha x(W_{N,\ell})+\sum_{i\in N\cup\{k\}}b_iy_i=a_0$. After plugging in the characteristic vector of P'_k into this equation, it follows that $a_0+b_i+b_\ell=a_0$, and therefore we have $b_i=-b_\ell=\beta$, for all $i\in N$. Equation (16) becomes now $a_0x(\overline{W}_{N,\ell})+\alpha x(W_{N,\ell})+\beta y(N)-\beta y_\ell=a_0$. Notice that solution $\{\ell\}$ also belongs to $F(\ell,N)$, which implies that $\alpha-\beta=a_0$. Finally, substituting a_0 in the previous equation, and using the equation (4), $x(V)=1$, we end up with the following form of (16):

$$\beta[-x(\overline{W}_{N,\ell})+y(N)-y_\ell=-1],$$

which together with equation (4) concludes the proof. □

5 Computational Results

For testing the computational performance of the presented formulations we have considered both directed and undirected MWCS instances. The $(CYCLE)$ model of Backes et al. [1] has been developed for directed graphs (regulatory networks) with K-cardinality constraints, i.e., any feasible solution has to be comprised by exactly K nodes (for a given $K > 1$). Executables of this implementation are available online (see [12]). For the $(PCStT)$ and (CUT) models we have developed our own B&C implementations that work with and without cardinality constraints. The real-world instances used in [1] require K-cardinality constraints. Therefore, in the part of our computational study conducted on digraphs, we impose cardinality constraints for all three models, $(PCStT)$, (CUT) and $(CYCLE)$. For the other set of instances we take the size of the unconstrained optimal solution (obtained by the (CUT) model) and provide the corresponding value of K as input to the $(CYCLE)$ model.

In the following, we describe (i) components of the designed B&C algorithms and some implementation details, (ii) a testbed used for the experiments, and (iii) an extensive analysis of the obtained results.

5.1 Branch-and-Cut Algorithms

Separation of Inequalities: For the $(PCStT)$ model, connectivity inequalities (2) are separated within the B&C framework by means of the maximum flow algorithm given by [5]. The separation problem is solved on a support graph whose arc capacities are given by the current LP value of $\mathbf{z}$ variables. We randomly select a terminal $v \in V$ such that $p_v > 0$ and $y_v > 0$, and calculate the maximum flow between the artificial root and v, and insert the corresponding constraint (2), if violated.

For the (CUT) formulation, the separation of (gNSep) is performed by solving the maximum flow problems as described in the proof of Proposition 2, with arc capacities given by (10).

In all cases, instead of adding a single violated cut per iteration, we use *nested*, *back-flow* and *minimum cardinality* cuts (see also [17, 20]) to add as many violated cuts as possible. We restrict the number of inserted cuts within each separation callback to 25.

Primal Heuristic: Our primal heuristic finds feasible solutions using the information available from the current LP solution in a given node of the branch-and-bound tree. Although we develop two different B&C algorithms, derived from two MIP models, the embedded primal heuristics are based on the same idea. We select a subset of potential "key-players" (nodes with a positive outgoing degree and with sufficiently large $\mathbf{y}$ values) and run a restricted breadth-first search (BFS) from each of them. Out of the constructed connected components, i.e., feasible solutions of the MWCS, we select the one with the largest total weight.

MIP Initialization: We initialize the $(PCStT)$ model with the root out-degree constraints (3). For the undirected MWCS, we also add symmetry-breaking constraints (similar to (14)) and inequalities $z_{ji} + z_{ij} \leq y_i$, for all $e : \{i, j\} \in E$ since they avoid too frequent calls of the maximum flow procedure. For the variants where no cardinality constraint is defined, we also include the flow-balance constraints: $z(\delta^-(i)) \leq z(\delta^+(i))$, for all $i \in V$ such that $p_i \leq 0$. These constraints ensure that a node with non-positive weight can not be a leaf in an optimal PCStT solution.

We initialize the (CUT) model with the constraints (4), (5), (6). For the cases where no cardinality constraint is imposed, the out-degree constraints (13) are also included. Finally, the symmetry-breaking constraints (14) are added for the undirected case.

Implementation: The proposed approaches were implemented using CPLEX^{TM}12.3 and Concert Technology. All CPLEX parameters were set to their default values, except the following ones: (i) CPLEX cuts were turned off, (ii) CPLEX heuristics were turned off, (iii) CPLEX preprocessing was turned off, (iv) the time limit was set to 1800 seconds (except for the instances from [1]), and (v) higher branching priorities were given to **y** variables, in the case of the $(PCStT)$ models, and to **x** variables, in the case of the (CUT) model. All the experiments were performed on a Intel Core2 Quad 2.33 GHz machine with 3.25 GB RAM, where each run was performed on a single processor.

5.2 Benchmark Instances

We have considered two sets of benchmark instances arising from applications in systems biology and from network design.

System Biology Instances: We have considered instances used in [8] and [1]. In [8], only a single protein-protein interaction network is considered. The instance is presented as an undirected graph comprised by 2034 nodes (proteins) and 8399 edges (interactions). The considered protein-protein interaction network corresponds to a well studied human one and the protein scores come from a lymphoma microarray dataset (LYMPH). The instance is available at [21].

In [1], six instances of regulatory networks, i.e., directed graphs, were considered. These instances have the same underlying network (KEGG human regulatory network of protein complexes), which is a graph comprised by 3917 nodes and 133 310 arcs. The differences between the six benchmark instances of this set are the scores associated to the proteins (or protein complexes) which depend on the pathogenic process under consideration. All the instances are available online (see [12]). For providing a valid comparison with the method proposed in [1], it is necessary to impose cardinality constraints to the solutions. Values $K \in \{10, 11, \ldots, 25\}$ are considered. This leads to 16 different instances for each of

the six different score settings.

Network Design Instances: These are Euclidean random instances which are generated as proposed by Johnson, Minkoff, and Phillips in their paper on the Prize-Collecting Steiner Tree Problem [16]. The topology of these instances is similar to street networks. First, n nodes are randomly located in a unit Euclidean square. A link between two nodes i and j is established if the Euclidean distance d_{ij} between them is no more than $\alpha/\sqrt{n}$, for a fixed $\alpha > 0$.

To generate node weights, we performed the following procedure: δ% of the nodes are randomly selected to be associated with non-zero weights. Out of them, ε% are associated with a weight taken uniformly randomly from $[-10,0]$ and the remaining ones are associated with a weight taken uniformly randomly from $[0,10]$.

When generating these instances we do not impose whether links are directed or not. When reading the input files we define if the link between i and j corresponds to an edge $e : \{i, j\}$ or to an arc $a : (i, j)$. This allows us to use the same set of instances for both, the directed and the undirected case.

For the computational experiments we considered $n \in \{500, 750, 1000, 1500\}$, $\alpha \in \{0.6, 1.0\}$, $\delta \in \{0.25, 0.50, 0.75\}$, $\varepsilon \in \{0.25, 0.50, 0.75\}$. This leads to 18 instances for each fixed value of n.

5.3 Algorithmic Performance

MWCS on Digraphs: For this study, we consider the instances GSE13671, GDS1815, HT-29-8, HT-29-24, HT-116-8, HT-116-24 from [1] and our randomly generated instances.

In Fig. 3, using the box plots we show the $\log_{10}$-values of the running times for the three approaches considering all instances of [1] and all values of K. There are $16 \times 6 = 96$ problems in total for each approach. The values marked with an asterisk correspond to the $\log_{10}$-values of the mean running time (shown as the label next to the asterisk). The values marked with symbol $\times$ correspond to the $\log_{10}$-values of the maximum running times (the label next to it shows the name of the instance, K, and the running time). The obtained results indicate that, for this group of instances, $(PCStT)$ is the approach with the worst performance since most of the running times are at least one order of magnitude larger than the ones of the other two approaches. When comparing (CUT) and $(CYCLE)$, one can observe that the distribution of the running times of the $(CYCLE)$ model has a larger dispersion (the *box* is wider) and its outliers are almost one order of magnitude larger than the maximum running times of the (CUT) model. In a few cases however the $(CYCLE)$ model solves some instances faster than the (CUT) model (which can be seen from the minimum values and the values in the first-quartile). Overall, the mean value of the running times of the (CUT) model is 22 sec which is almost three times smaller than the mean running time of the $(CYCLE)$ model (77 sec). The value of the maximum running time of the (CUT) model is 193 sec which is more than 10 times smaller than the

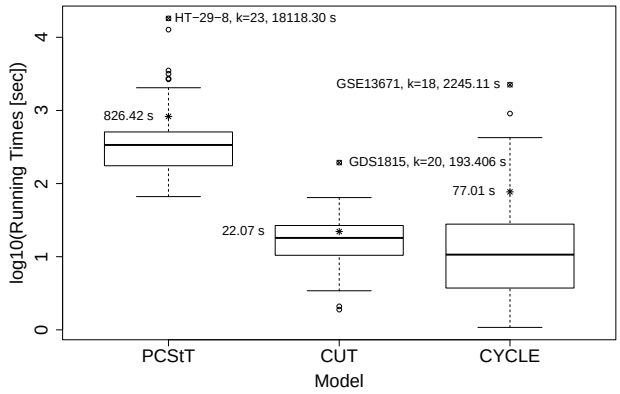


Fig. 3 Box plots of $\log_{10}$-values of the running times [sec] (instances from [1], $K \in \{10,\ldots,25\}$).

maximum running time of the $(CYCLE)$ model (2245 sec, reached for $K = 18$ for the instance GSE13671, see Fig. 3). The fact that the box of the (CUT) model is considerably narrower than the box of the $(CYCLE)$ model, indicates that the (CUT) approach is more robust regarding the variation of the scores of protein complexes and the value of K.

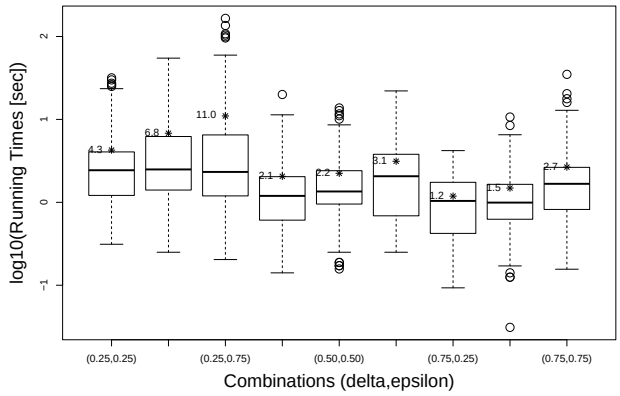
In Table 1 we report for each instance from [1] the average values (over all $K \in \{10,\ldots,25\}$) of the running times and the average number of cuts added for each of the $(PCStT)$, (CUT) and $(CYCLE)$ models (cf. columns Time(sec), #(2), #(gNSep) and #(7), respectively). In column δ we show the fraction of nodes with a score different than 0 and in column ε the fraction of them with a negative score. The results indicate that the performance of the $(CYCLE)$ model strongly depends on the instances under consideration (the average running times of GSE13671 are two orders of magnitude larger that the ones of HT-116-8), which also explains the dispersion shown in Fig. 3. Likewise, for the $(PCStT)$ model, the average running time for the instance HT-29-8 is an order of magnitude larger than for the instance GSE13671. In contrast to the unstable performance of $(PCStT)$ and $(CYCLE)$ models, the (CUT) model seems to be more independent on the type of considered instances. From the same table we may conclude that the number of cuts needed to prove the optimality is one order of magnitude smaller for the (CUT) model than for the other two models. This means that the (gNSep) cuts are more effective in closing the gap than the (7) and (2) cuts. Regarding δ and ε, it seems that the (CUT) model is not sensitive to their values, while the $(CYCLE)$ model performs better when ε is smaller.

For the set of Euclidean network instances, running times of the (CUT) and $(CYCLE)$ model are given in Fig. 4(a) and 4(b), respectively (for many instances we reached the time-limit for the $(PCStT)$ model, so we do not consider it here). This time we group instances according to different combinations of (δ,ε) values. Each box contains $16 \times 8 = 128$ values obtained for the settings: $K \in \{10,\ldots,25\}$,

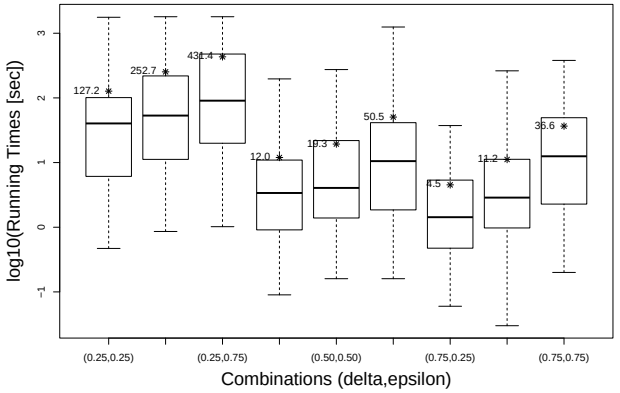
$n \in \{500, 750, 1000, 1500\}$ and $\alpha \in \{0.6, 1.0\}$. Comparing Fig. 4(a) and 4(b) we observe that although the average running times (marked with asterisk) of the (CUT) model are in general one order of magnitude smaller than those of the $(CYCLE)$ model, both of them present a similar pattern: (i) For a given δ, the increase of ε from 0.25 to 0.75 produces a worsening of the algorithmic performance. This worsening is visible not only in the increase of the running times, but also in their higher dispersion (wider boxes and more outliers). Increasing ε (for a fixed δ), means that a larger proportion of nodes has a negative weight; since our goal is to find a connected component of exactly K nodes the more nodes with negative weight, the more difficult is the task of reaching the "attractive" nodes that lead to a better solution. (ii) On the other hand, increasing δ from 0.25 to 0.75 produces an improvement of the algorithmic performance, i.e., the more nodes with non-zero weights, the easier the problems. One possible reason for this could be the symmetries induced by a large portion of nodes with zero weight (as it is the case for $\delta = 0.25$). Hence, by decreasing this portion (i.e., increasing δ) the cutting-planes that are added through the separation become more effective, and the primal heuristic is able to find more diverse, and eventually better, incumbent solutions.

MWCS on Undirected Graphs: For this computational comparison we do not impose cardinality constraints. In order to be able to perform a comparison with the $(CYCLE)$ model that requires a digraph G and K as its input, we run the $(CYCLE)$ model with (i) G transformed into a digraph, and (ii) with the value of K set to be the size of the optimal unconstrained MWCS solution (obtained by, e.g., the (CUT) model). For these graphs we impose a time limit of 1800 seconds. Fig. 5 shows the performance profile of the three approaches regarding the total running time. Fig. 6 shows the performance profile of the achieved gaps within this time limit. We observe that also in the case of undirected graphs, the (CUT) approach significantly outperforms the $(CYCLE)$ and the $(PCStT)$ approach: While the (CUT) approach produces solutions of less than 1% of gap in almost 100% of the instances, the $(PCStT)$ approach produces solutions with more than 15% of gap in more than 40% of the instances. The $(CYCLE)$ approach solves about 50% of instances to optimality, with most of the gaps of the unsolved instances being below 15%.

In Table 2 we provide more details on these results. Each row corresponds to a fixed value of n, with 18 different instances obtained by varying δ, ε and α. Column #NOpt indicates how many out of those 18 instances were not solved to optimality within the imposed time limit of 1800 seconds. For a given n, and for each of the three approaches we additionally report on the following values: the average running time (cf. column Time(sec)); the average gap of those instances that were not solved to optimality (cf. column Gap(%)), and the average number of inserted cutting planes (cf. columns #(2), #(gNSep), #(7), respectively). These results show that the (CUT) model is by far more effective than the $(CYCLE)$ model for this group of instances. The average running times of the (CUT) model are one order of magnitude smaller than those of the $(PCStT)$ and $(CYCLE)$ model. All but four instances can be solved by the (CUT) model to optimality, while in the case of the



(a) Influence of δ and ε on the performance of the (CUT) model (random instances, $K \in \{10,\ldots,25\}$).



(b) Influence of δ and ε on the performance of the $(CYCLE)$ model (random instances, $K \in \{10,\ldots,25\}$).

Fig. 4 Dependance of the running times on the (δ,ε) settings.

$(CYCLE)$ and $(PCStT)$ model, 29 and 42 instances remain unsolved, respectively. The number of cutting planes of type (gNSep) needed to close the gap is one order magnitude smaller than the number of cuts of type (7) or (2).

So far, it seems clear that for the considered instances the (CUT) model significantly outperforms the $(PCStT)$ approach. However for the LYMPH instance studied in [8], for which $\delta = 1.0$ and $\varepsilon = 0.97$, the $(PCStT)$ model takes only 3.19 seconds to find the optimal solution while the $(CYCLE)$ model takes 15.56 seconds, and the (CUT) model 50.70 seconds. The optimal solution, whose objective value is 70.2, is comprised by 37 nodes with positive weight and 9 with negative weight. It is not easy to derive a concrete answer of why, for this particular instance, the

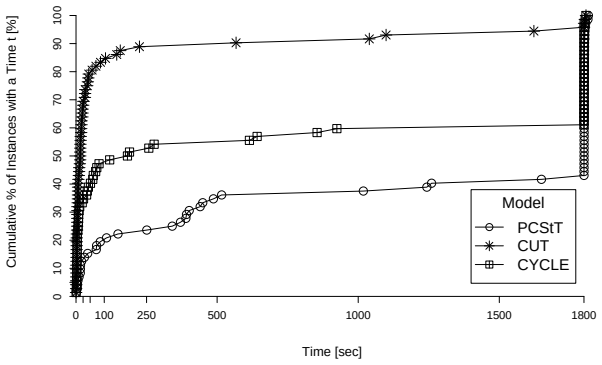


Fig. 5 Performance profile of running times on random undirected instances.

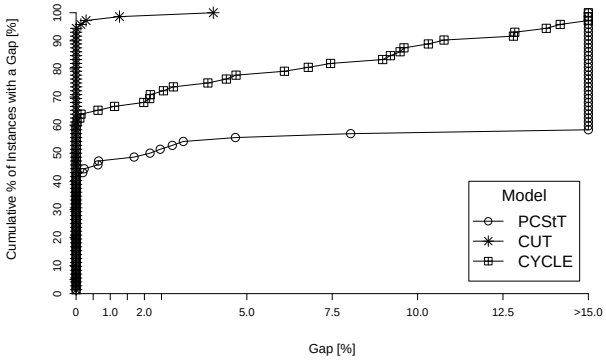


Fig. 6 Performance profile of final gaps (%) on random undirected instances.

$(PCStT)$ model is faster than the (CUT) model. The following two factors could be responsible for this behavior: (i) the sparsity of the graph (the number of edges is approximately four times the number of nodes, while in random instances this ratio is almost 10) which means that the number of $\mathbf{z}$ variables is not too large, and (ii) there are significantly less symmetries due to the fact that there are no nodes with zero weight. These factors might explain why, in this particular case, it becomes easier to solve the problem with the prize-collecting Steiner tree reformulation, rather than directly looking for a connected component that maximizes the objective function.

6 Conclusion

Our work was motivated by the wide range of applications of the MWCS and a recent work of Backes et al. [1] who were the first ones to propose a MIP model for the MWCS derived on the set of node variables only. In this paper we were able to provide a tight MIP model that outperforms the model from [1] both theoretically and computationally. The new model also works on the space of node variables and is valid for all previously studied variants of the MWCS (cardinality constrained, budget constrained and undirected/directed one). We have studied the CS polytope and we have shown that the newly introduced family of generalized node-separator inequalities is facet defining. Our computational study has shown that the new approach outperforms the previously proposed ones, in particular if the inputs are digraphs with non-empty subsets of zero-weight nodes.

Acknowledgements We are deeply thankful to Christina Backes from the Department of Human Genetics, Saarland University, who helped in the understanding and interpretation of the regulatory network instances considered in this paper. This research is partially conducted during the research stay of Ivana Ljubić at the TU Dortmund, supported by the APART Fellowship of the Austrian Academy of Sciences. This support is greatly acknowledged. Eduardo Álvarez-Miranda thanks the Institute of Advanced Studies of the Università di Bologna from where he is PhD a Fellow.

Table 1 Average values for instances from [1] ($K \in \{10, \ldots, 25\}$).

			(*PCStT*)		(*CUT*)		(*CYCLE*)	
Instance	δ	ε	Time(sec)	#(2)	Time(sec)	#(gNSep)	Time(sec)	#(7)
GSE13671	0.89	0.73	176.11	1206	17.85	97	341.95	3754
GDS1815	0.92	0.64	878.63	3565	46.09	225	37.95	1264
HT-29-8	0.92	0.66	2846.36	5400	22.03	182	14.17	178
HT-29-24	0.92	0.61	196.56	1292	11.40	61	60.59	1330
HT-116-8	0.92	0.54	623.10	2214	15.26	108	3.21	129
HT-116-24	0.92	0.55	237.78	1149	19.82	93	4.19	130
Average			826.42	2471	22.07	128	77.01	1131

Table 2 Average values for different values of n (random instances, $\alpha \in \{0.6, 1.0\}$, $\delta, \varepsilon \in \{0.25, 0.50, 0.75\}$, 18 problems per each n).

		(*PCStT*)				(*CUT*)				(*CYCLE*)			
#nodes	#arcs	Time(sec)	Gap(%)	#(2)	#NOpt	Time(sec)	Gap(%)	#(gNSep)	#NOpt	Time(sec)	Gap(%)	#(7)	#NOpt
500	4558	677.24	>15.00	1055	5	15.30	–	69	0	615.36	5.50	4289	6
750	7021	1243.57	>15.00	1552	11	108.78	1.27	99	1	471.68	2.64	1721	4
1000	9108	1304.76	>15.00	1955	12	150.03	0.29	201	1	990.84	6.76	3176	9
1500	14095	1526.41	>15.00	2021	14	453.82	2.08	373	2	1086.19	10.55	2139	10

References

1. Backes, C., Rurainski, A., Klau, G., Müller, O., Stöckel, D., Gerasch, A., Küntzer, J., Maisel, D., Ludwig, N., Hein, M., Keller, A., Burtscher, H., Kaufmann, M., Meese, E., Lenhof, H.: An integer linear programming approach for finding deregulated subgraphs in regulatory networks. Nucleic Acids Research **1**, 1–13 (2011)
2. Bateni, M., Chekuri, C., Ene, A., Hajiaghayi, M., Korula, N., Marx, D.: Prize-collecting Steiner problems on planar graphs. In: D. Randall (ed.) Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011, pp. 1028–1049. SIAM (2011)
3. Carvajal, R., Constantino, M., Goycoolea, M., Vielma, J., Weintraub, A.: Imposing connectivity constraints in forest planning models (2011). Submitted for publication
4. Chen, C.Y., Grauman, K.: Efficient activity detection with max-subgraph search. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012 (CVPR), pp. 1274–1281. IEEE (2012)
5. Cherkassky, B.V., Goldberg, A.V.: On implementing push-relabel method for the maximum flow problem. Algorithmica **19**, 390–410 (1994)
6. Chimani, M., Kandyba, M., Ljubic, I., Mutzel, P.: Obtaining optimal *k*-cardinality trees fast. ACM Journal of Experimental Algorithmics **14** (2009)
7. Dilkina, B., Gomes, C.: Solving connected subgraph problems in wildlife conservation. In: A. Lodi, M. Milano, P. Toth (eds.) CPAIOR, *LNCS*, vol. 6140, pp. 102–116. Springer (2010)
8. Dittrich, M., Klau, G., Rosenwald, A., Dandekar, T., Müller, T.: Indetifying functional modules in protein-protein interaction networks: an integrated exact approach. Bioinformatics **24**, i223–i231 (2008)
9. Feigenbaum, J., Papadimitriou, C.H., Shenker, S.: Sharing the cost of multicast transmissions. J. Comput. Syst. Sci. **63**(1), 21–41 (2001)
10. Fischetti, M., Hamacher, H.W., Jørnsten, K., Maffioli, F.: Weighted k-cardinality trees: Complexity and polyhedral structure. Networks **24**(1), 11–21 (1994)
11. Fügenschuh, A., Fügenschuh, M.: Integer linear programming models for topology optimization in sheet metal design. Mathematical Methods of Operations Research **68**(2), 313–331 (2008)
12. (September 10th 2012). Url = http://genetrail.bioinf.uni-sb.de/ilp/
13. Goldschmidt, O., Hochbaum, D.S.: *k*-edge subgraph problems. Discrete Applied Mathematics **74**(2), 159–169 (1997)
14. Hochbaum, D.S., Pathria, A.: Node-optimal connected k-subgraphs (1994). Manuscript, UC Berkeley
15. Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.: Discovering regulatory and signalling circuits in molecular interaction networks. Bioinformatics **18 (Suppl. 1)**, s233–s240 (2002)
16. Johnson, D.S., Minkoff, M., Phillips, S.: The prize-collecting Steiner tree problem: Theory and practice. Proc. 11th ACM-SIAM Symp. Discrete Algorithms, 9-11 January 2000, San Francisco, USA **SODA 2000**, 760–769 (2000)
17. Koch, T., Martin, A.: Solving Steiner tree problems in graphs to optimality. Networks **32**, 207–232 (1998)
18. Lee, H., Dooly, D.: Decomposition algorithms for the maximum-weight connected graph problem. Naval Research Logistics **45**, 817–837 (1998)
19. Lee, H., Dooly, D.R.: Algorithms for the constrained maximum-weight connected graph problem. Naval Research Logistics **43**, 985–1008 (1996)
20. Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G., Mutzel, P., Fischetti, M.: An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. Mathematical Programming **Series B 105**, 427–449 (2006)
21. (September 10th 2012). Url = http://www.planet-lisa.net/
22. Yamamoto, T., Bannai, H., Nagasaki, M., Miyano, S.: Better decomposition heuristics for the maximum-weight connected graph problem using betweenness centrality. In: J. Gama, V. Costa, A. Jorge, P. Brazdil (eds.) Discovery Science, *Lecture Notes in Computer Science*, vol. 5808, pp. 465–472. Springer (2009)