Als Manuskript gedruckt

Technische Universität Dresden
Herausgeber: Der Rektor

Integer linear programming formulations for the minimum connectivity inference problem and model reduction principles

Muhammad Abid Dar, Andreas Fischer, John Martinovic, and Guntram Scheithauer

Preprint MATH-NM-03-2019

September 2019

Integer linear programming formulations for the minimum connectivity inference problem and model reduction principles

Muhammad Abid Dar[a], Andreas Fischer[a], John Martinovic[a,*], Guntram Scheithauer[a]

[a] *Institute of Numerical Mathematics, Technische Universität Dresden, 01062 Dresden, Germany*

Abstract

The minimum connectivity inference (MCI) problem represents an $\mathcal{NP}$-hard generalization of the well-known minimum spanning tree problem. Given a set of vertices and a finite collection of subsets (of this vertex set), the MCI problem requires to find an edge set of minimal cardinality so that the vertices of each subset are connected. Although the problem under consideration has appeared in different application-oriented scientific contexts in the last decades, (efficient) approaches to its exact solution have hardly been addressed in the literature. Currently, even the most promising ILP formulation (an improved flow-based model) can only deal with rather small instances in reasonable times. In order to also tackle practically relevant problem sizes, our contribution is twofold: At first, we propose several new modelling frameworks for the MCI problem and investigate their theoretical properties as well as their computational behavior. Moreover, we introduce the concepts of simple model reduction and generalized model reduction which can be applied to reduce the numbers of variables and/or constraints in the various formulations. Based on extensive numerical experiments, the practical advantages of these principles are validated.

Keywords: Minimum Connectivity Inference, Minimum Spanning Tree, MILP, Model Reduction, Polytopes

1. Introduction

Let us consider an undirected complete graph and a finite collection of subsets (hereinafter referred to as *clusters*) of its vertex set. Then, the Minimum Connectivity Inference (MCI) problem is to find an edge set with minimum cardinality which contains a spanning tree for every cluster. This problem can be seen as a generalization of the well-known Minimum Spanning Tree (MST) problem. However, while the MST problem is solvable in polynomial time, the MCI problem is known to be $\mathcal{NP}$-hard [6, 13].

In the literature, several applications of the MCI problem can be encountered, but – depending on the particular context – different names for this optimization problem have been established. The problem itself was first described in [12], with reference to a publicly not available article from 1976, for the placement of valves in vacuum systems. Closely related

*Corresponding author

Email addresses: muhammad_abid.dar@tu-dresden.de (Muhammad Abid Dar), Andreas.Fischer@tu-dresden.de (Andreas Fischer), john.martinovic@tu-dresden.de (John Martinovic), Guntram.Scheithauer@tu-dresden.de (Guntram Scheithauer)

to this, the name Subset Interconnection Design problem was proposed in [13]. In recent years, other applications (and names) mainly originated from different network design problems [3, 4, 5, 6, 15, 17] and structural biology [1, 2]. For the sake of simplicity, here we would only like to highlight the fact that, in the latter references, the name MCI problem was introduced, whereas for a more detailed discussion of the various terminologies, we refer the reader to the introductory section of [11].

Most of the papers already cited as well as further contributions like [14, 22] either deal with complexity issues or heuristic methods related to the (solution of the) MCI problem. Contrary to that, so far, not many efforts have been conducted to compute an exact solution of the MCI problem. To the best of our knowledge, the first attempt in this regard was made in [1] by presenting a flow-based mixed interger linear programming (MILP) formulation. However, given the large number of binary variables and the rather poor quality of the related LP bound, only instances of small sizes can be solved using this formulation. Based on this preliminary research, the authors of the current work recently suggested an improved MILP formulation, see [11], having

- fewer numbers of variables and constraints,
- a stronger LP relaxation (thanks to the addition of valid inequalities),
- a more favorable general structure (due to the elimination of linear dependencies among the flow propagation constraints).

This new model together with instance reduction techniques (introduced in [5] and [11], and thoroughly evaluated in [10]) enable to cope with larger, but still rather medium-sized, instances of the MCI problem.

To further enlarge the range of MCI instances that can be solved to proven optimality in reasonable amounts of time, new modeling frameworks and reductions are presented in this paper. More precisely, the main contributions of this article are twofold:

i) First of all, we exploit the relationship between the MCI problem and the MST problem to derive new (M)ILP formulations for the problem under consideration. Besides discussing the differences in terms of the numbers of variables and constraints, we also focus on the strength of the linear relaxations (of the various approaches) by investigating the subset relations of the corresponding polytopes from a theoretical point of view.

ii) Secondly, we establish two types of *model reductions* (a simple version and a generalized version) to further decrease the numbers of variables and constraints in the (M)ILP formulations. It is worth mentioning that these reductions are introduced in an abstract and generic way so that they are applicable to all considered approaches.

The computational performance of these models and the effects of model reduction techniques are compared by extensive numerical experiments. Based on a wide variety of differently characterized (randomly generated) instances, we clearly point out the advantages and disadvantages of the new formulations, also showing that any approach is competitive for at least one major group of the attempted instances.

The paper is organized as follows: In Section 2, the formal definition of the MCI problem is presented along with some required notations. For the sake of a better comprehension, we

will then recapitulate five different formulations of the MST problem (known in the literature) and discuss the relationship between the respective polytopes in Section 3. Based on this preliminary study of different approaches for the MST problem, Section 4 introduces the resulting (M)ILP models of the MCI problem and also includes the theoretical relationships between the corresponding polytopes of the LP relaxations. Afterwards, the new idea of model reductions is derived in two parts. In the first part, a *simple model reduction* technique is introduced in Section 5. The second part in Section 6 then generalizes this reduction idea. Our computational results are shown and discussed in Section 7 and they are complemented by some conclusions and an outlook in Section 8.

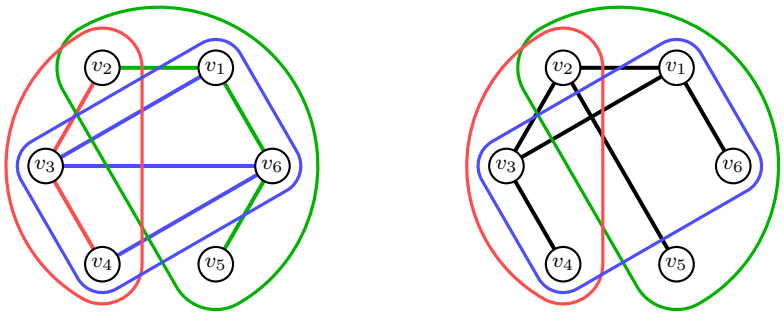
2. Preliminaries

Let $G = (V, E)$ be a simple, undirected, and complete graph with vertex set $V = \{1, 2 \ldots, m\}$ and edge set $E = \{e = \{j, k\} \mid j, k \in V, j \neq k\}$. A *tree* of the graph $G = (V, E)$ is a connected subgraph $T = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$ containing no cycles. Sometimes, we briefly say that T *spans* the vertices of V'. Moreover, T is said to be a *spanning tree* of G if T spans V, that is $V' = V$. For a graph $G' = (V, E')$ with $E' \subseteq E$ and a subset $V_0 \subseteq V$, the *induced subgraph* $G'[V_0]$ represents the graph with vertex set V_0 and all those edges of E' whose both vertices belong to V_0.

Definition 1. Let

$$\mathcal{C} := \left\{ V_i \subseteq V \;\middle|\; \bigcup_{i \in I} V_i = V,\, 1 < |V_i|, i \in I := \{1, \ldots, n\} \right\}$$

denote a finite collection of subsets of V. Then, the *Minimum Connectivity Inference* (MCI) problem is to determine a set $E^\star \subseteq E$ of minimum cardinality, such that the induced subgraphs $G^\star[V_i]$ of $G^\star = (V, E^\star)$ are connected for all $i \in I$.



(a) A feasible solution (b) An optimal solution

Figure 1: An illustration of a feasible (1a) and an optimal (1b) solution of an MCI instance having six vertices $v_1, \ldots, v_6$. Moreover, there are three clusters represented by grouping their incident vertices together inside closed curves with different colors. The edges belonging to the (feasible) solution are drawn by thick lines.

The elements of the set $\mathcal{C}$ are typically called *clusters*, and the pair $(V, \mathcal{C})$ is termed as an *instance* of the MCI problem. Moreover, for a given instance $(V, \mathcal{C})$, $E^\star$ is called an *optimal*

edge set or a *(optimal) solution*, and the cardinal number $|E^\star|$ of $E^\star$ is the *optimal value* of this instance. An edge set $E' \subseteq E$ is said to be *feasible*, if the induced subgraphs $G'[V_i]$ of $G' = (V, E')$ are connected for all $i \in I$. By way of example, Figure 1 depicts a feasible and an optimal solution of a small MCI instance.

Since a (sub-)graph is connected if and only if it contains a spanning tree, the MCI problem is canonically related to spanning tree problems like the extensively studied minimum spanning tree problem. For a graph $G = (V, E)$ with given positive edge weights w_e, $e \in E$, the *minimum spanning tree* (MST) problem is to find a spanning tree of G with minimum sum of edge weights. It is well known that, as a consequence of matroid theory, this problem can already be solved by heuristic greedy approaches (like Kruskal's algorithm [18]) or a wide variety of ILP or MILP formulations, see [19] for a good overview. Moreover, MST problems naturally appear as subroutines in (approximation) algorithms for other problem classes, like the travelling salesman problem [7], clustering problems [24, 25, 26], or matching problems [23], thus leading to a wide range of further practical applications. In this article, we concentrate on five classes of (M)ILP formulations for the MST problem, all of which sharing the structural property that the same set of binary variables x_e, $e \in E$, is used (possibly together with additional variables) to model the choice of edges belonging to the spanning tree. Based on these considerations, corresponding MCI formulations will be derived afterwards.

Finally, the following notations are required in subsequent sections. In addition to (undirected) edges, sometimes also (directed) arcs are used. To this end, let A denote the set of all arcs[1] which can be assigned to graph $G = (V, E)$, i.e.,

$$A := \{(j, k) \mid j, k \in V, j \neq k\}.$$

Furthermore, for any subset $S \subseteq V$, we define the arc sets

$$\begin{aligned} A(S) &:= \{(i, j) \in A \mid i \in S, j \in S\}, \\ A^+(S) &:= \{(i, j) \in A \mid i \in S, j \in V \setminus S\}, \\ A^-(S) &:= \{(i, j) \in A \mid i \in V \setminus S, j \in S\}, \end{aligned}$$

the edge set

$$E(S) := \{\{i, j\} \in E \mid i \in S, j \in S\},$$

and the set $\Pi(S)$ containing all partitions (of S). A *partition* $\mathcal{P} = \{\rho_1, \dots \rho_k\}$ of $S \subseteq V$ is a finite collection $\rho_1, \dots, \rho_k \subseteq S$ of subsets of S, so that $\emptyset \notin \mathcal{P}$, $\bigcup_{i=1}^{k} \rho_i = S$, and $\rho_i \cap \rho_j = \emptyset$ for $i \neq j \in \{1, \dots, k\}$ are satisfied. For any partition $\mathcal{P} \in \Pi(S)$, we further define the set

$$\delta(\mathcal{P}) := \{e = \{u, v\} \mid u \in \rho_i, v \in \rho_j, i, j \in \{1, \dots, k\} \text{ with } i \neq j\}$$

containing all edges having their vertices in different partition classes. Finally, we would like to mention that, for the sake of simplicity, we use $A^-(i)$ and $A^+(i)$ instead of $A^-(\{i\})$ and $A^+(\{i\})$, and shortly write $\delta(S)$ for $\delta(\{S, V \setminus S\})$.

[1] Note that, for the sake of simplicity, we will refer to both graphs (the directed and the undirected version) by G. However, whenever the particular interpretation of this symbol is not clear (from the context itself) at a specific position within the paper, we recall whether an orientation of the edges is assumed or not.

3. ILP and MILP formulations of the MST problem

Over the last couple of decades, a large number of different (M)ILP formulations for the MST problem was proposed and discussed in the literature. In this section, we present a selection of the most important MST models, which are later used to define corresponding formulations of the MCI problem in Section 4. While our repetition is intended to only briefly illustrate the various well-known solution approaches, we refer the reader to the survey presented in [19] or the articles [9, 16, 20] for a more detailed overview.

For a given vector w of positive edge weights, the considered MST models possess the following common structure

$$\text{minimize } w^\top x \quad \text{s.t.} \quad x \in P_{\mathcal{M}} \cap \{0,1\}^{|E|}, \tag{3.1}$$

where the symbol $\mathcal{M}$ indicates one of the respective MST models, and $P_{\mathcal{M}} \subset [0,1]^{|E|}$ denotes the projection of all feasible solutions of the LP relaxation onto $[0,1]^{|E|}$, i.e., $P_{\mathcal{M}}$ is a polytope and represents the feasible region of the x-variables for the LP relaxation of the corresponding MST model $\mathcal{M}$. Additionally, we use $\mathcal{T}$ to denote the set of the incidence vectors of all spanning trees related to $G = (V, E)$, i.e.,

$$\mathcal{T} := \{x \in \{0,1\}^{|E|} \mid E(x) \text{ defines a spanning tree of } G = (V, E)\},$$

where $E(x) := \{e \in E \mid x_e = 1\}$. Note that, for the sake of simplicity, we just provide the definition of the polytope $P_{\mathcal{M}}$ in the description of the MST models mentioned below.

3.1. Subtour elimination/packing formulation ($\mathcal{M} = sub$)

This MST formulation is based on the property that a spanning tree of a graph having $m = |V|$ vertices defines an acyclic subgraph with $m - 1$ edges. The related polytope P_{sub} is given by

$$P_{sub} := \{x \in [0,1]^{|E|} \mid (3.2) \text{ and } (3.3) \text{ are fulfilled}\},$$

where

$$\sum_{e \in E} x_e = |V| - 1, \tag{3.2}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \qquad \text{for all } S \subset V \text{ with } S \neq \emptyset. \tag{3.3}$$

Constraint (3.3) is known as a *packing constraint* or *subtour elimination constraint*. It guarantees that no cycle is contained in any $x \in P_{sub} \cap \{0,1\}^{|E|}$. Moreover, the *cardinality constraint* (3.2) formulates the well known fact that any spanning tree of $G = (V, E)$ contains exactly $|V| - 1$ edges.

Proposition 1 ([19]). *The extreme points of the polytope P_{sub} belong to the set $\mathcal{T}$ and $P_{sub} = \text{conv}(\mathcal{T})$ holds, where $\text{conv}(\cdot)$ denotes the convex hull.*

3.2. Cutset formulation ($\mathcal{M} = cut$)

In addition to (3.2), the cutset formulation rather focusses on the connectivity of a spanning tree. The corresponding polytope is given by

$$P_{cut} := \{x \in [0,1]^{|E|} \mid (3.2) \text{ and } (3.4) \text{ are fulfilled}\},$$

where

$$\sum_{e \in \delta(S)} x_e \geq 1, \qquad \text{for all } S \subset V \text{ with } S \neq \emptyset. \tag{3.4}$$

Constraint (3.4) ensures that, for any $x \in P_{cut} \cap \{0,1\}^{|E|}$, the edges in $E(x)$ connect any nonempty vertex set $S \subset V$ with the complementary vertex set $V \setminus S$. In fact, (3.4) contains $2^{m-1} - 1$ inequalities.

Proposition 2 ([19]). *It holds that $P_{sub} \subseteq P_{cut}$, where equality is not true, in general. Moreover, P_{cut} can have fractional extreme points.*

A direct consequence of this result is that, in the general case, there is no equality between $\text{conv}(\mathcal{T})$ and P_{cut}.

3.3. Multi-cut/partition-based formulation ($\mathcal{M} = mcut$)

In comparison to the cutset formulation, a stronger formulation (in the sense of a smaller polytope) can be obtained if all partitions of V are employed instead of only considering the bisections. The resulting polytope is

$$P_{mcut} := \{x \in [0,1]^{|E|} \mid (3.2) \text{ and } (3.5) \text{ are fulfilled}\},$$

where

$$\sum_{e \in \delta(\mathcal{P})} x_e \geq |\mathcal{P}| - 1, \qquad \text{for all } \mathcal{P} \in \Pi(V). \tag{3.5}$$

It is obvious that $P_{mcut} \subseteq P_{cut}$ holds, since (3.5) is a superset of the inequalities appearing in (3.4).

Proposition 3 ([19, 21]). *It holds that $P_{mcut} = P_{sub} = \text{conv}(\mathcal{T})$.*

Note that, in [21], this result was even shown for a more general version of the MST problem, called the *generalized minimum spanning tree problem*. Whereas the polytopes P_{sub}, P_{cut}, and P_{mcut} are subsets of $\mathbb{R}^{|E|}$ with $|E| = \frac{1}{2}m(m-1)$, their descriptions rely on an exponential number of constraints (in terms of m). In contrast to this, the following two polytopes are based on a polynomial number of constraints.

3.4. Flow-based formulation ($\mathcal{M} = flow$)

Another classical approach to model the connectivity within a graph is based on flow propagation. To this end, here we consider a flow-based approach appearing in [16]. Please note that, in the literature, further closely related variants of this formulation can be found, see [8] or [19] by way of example.

In this flow-based formulation, from each vertex but one (say vertex s) exactly one unit

of flow has to be sent to the sink s. Hence, altogether $|V|-1$ units of flow (emanating from the vertices $v \in V \setminus \{s\}$) have to be collected in the sink. In this interpretation, an edge belongs to a (feasible) solution if and only if it is required to propagate any flow. In addition to the x_e-variables, $e \in E$, further variables are needed to model the flow. Since, for each edge $e = \{u, v\}$, a flow is possible in both directions u to v and v to u, these new variables have to be related to an orientation. To this end, we introduce $f_{(u,v)}$ and $f_{(v,u)}$ to indicate the total flow on edge $e = \{u, v\}$ in the direction u to v and v to u, respectively.

Within this paper, we use the polytope

$$P_{flow} := \{x \in [0,1]^{|E|} : \exists\, f = (f_a)_{a \in A} \text{ such that } (x, f) \text{ fulfills (3.2) and (3.6) - (3.8)}\}$$

of the following flow-based formulation of the MST problem

$$\sum_{a \in A^-(v)} f_a - \sum_{a \in A^+(v)} f_a = \begin{cases} |V|-1, & \text{if } v = s, \\ -1, & \text{otherwise,} \end{cases} \qquad v \in V, \tag{3.6}$$

$$f_{(u,v)} + f_{(v,u)} \leq (|V|-1)x_e, \qquad e = \{u, v\} \in E, \tag{3.7}$$

$$f_a \geq 0, \qquad a \in A. \tag{3.8}$$

Note that constraint (3.2) is induced by constraint (3.6) and, therefore, often not used in flow-based formulations in the literature. However, it restricts the set of feasible solutions of the LP relaxation and leads to a smaller polytope. Constraint (3.6) enforces the connectivity of the vertex set V, and constraint (3.7) translates the information of (3.6) from the f- to the x-variables. If there is any flow on edge $e = \{u, v\}$ (i.e., $f_{(u,v)} > 0$ or $f_{(v,u)} > 0$), then e has to belong to the solution (i.e., $x_e = 1$ has to be true), and if $e \in E$ does not appear in the solution (i.e., $x_e = 0$), then no flow is carried by any related arc. In the flow-based MILP model of the MST problem we have $|E|$ binary and $2|E|$ continuous variables. Besides the non-negativity constraint only $|V| + |E| + 1$ restrictions are involved.

Proposition 4 ([19]). *It holds that $P_{cut} \subseteq P_{flow}$.*

3.5. Martin's formulation ($\mathcal{M} = mar$)

A further formulation, focussing on the tree structure itself, was introduced in [20], and later it also appeared, with small modifications, in [9] and [16]. In the literature, this formulation is known as *Martin's* formulation, named after the author of [20]. Here, we summarize the version which was proposed in [16].

In this formulation, a new type of variables is used based on the following observation: if T is a spanning tree of $G = (V, E)$, then, for any vertex $w \in V$, an arborescence with root w is defined by assigning an appropriate direction to each edge of T. Therefore, for each edge $\{u, v\} \in E$ and each root vertex $w \in V \setminus \{u, v\}$, a decision variable $y_{uv,w}$ is defined where $y_{uv,w} = 1$ indicates that $\{u, v\}$ belongs to the spanning tree T and arc (v, u) is in the arborescence with root w, i.e., v is the predecessor of u. On the contrary, if $y_{uv,w} = 0$ holds, then v is not the direct predecessor of u or the edge $\{u, v\}$ does not belong to T. The corresponding polytope P_{mar} of Martin's formulation for the MST problem is given by

$$P_{mar} := \{x \in [0,1]^{|E|} : \exists\, y = (y_{a,w}) \text{ such that } (x, y) \text{ fulfills (3.2) and (3.9) - (3.11)}\},$$

where

$$\sum_{v \in V \setminus \{u,w\}} y_{uv,w} + x_e = 1, \qquad e = \{u,w\} \in E, \tag{3.9}$$

$$y_{uv,w} + y_{vu,w} = x_e, \qquad e = \{u,v\} \in E(V \setminus \{w\}), w \in V, \tag{3.10}$$

$$y_{uv,w} \in [0,1], \qquad \{u,v\} \in E(V \setminus \{w\}), w \in V. \tag{3.11}$$

Constraint (3.9) ensures that either an edge $e = \{u,w\}$ is in the spanning tree $T = \{e \in E : x_e = 1\}$, or any of the vertices $v \in V \setminus \{u,w\}$ is the predecessor of vertex u in the arborescence rooted at w (if $y_{uv,w} = 1$). Moreover, constraint (3.10) demands that if the edge $\{u,v\}$ (with $u,v \in V \setminus \{w\}$) is in T, then either v is the predecessor of u (if $y_{uv,w} = 1$) or u is the predecessor of v (if $y_{vu,w} = 1$) in the arborescence with root w. In Martin's formulation of the MST problem $O(m^2)$ binary variables as well as $O(m^3)$ continuous variables and constraints are contained. As shown in [20], Martin's formulation provides a representation of the convex hull $\text{conv}(\mathcal{T})$ of all spanning trees:

Proposition 5 ([20]). *It holds that* $P_{mar} = P_{sub} = \text{conv}(\mathcal{T})$.

4. ILP and MILP formulations of the MCI problem

Since the MCI problem can be seen as a generalization of the MST problem, we can apply all the MST problem formulations, presented in Section 3, to derive different (M)ILP models of the MCI problem. Hence, this section contains five models of the MCI problem (where four of them are new) and investigates relations between the various approaches. Since, in the MCI problem (similar to the MST problem), a set of edges has to be selected, we again use binary variables x_e, $e \in E$, to identify those edges which belong to a (feasible) solution. Moreover, we aim at formulating all the different models in a uniform manner as

$$\text{minimize } e^\top x \qquad \text{s.t.} \quad x \in \widetilde{P}_{\mathcal{M}} \cap \{0,1\}^{|E|}, \tag{4.1}$$

where $e = (1,\dots,1)^\top$ is of appropriate size, and $\widetilde{P}_{\mathcal{M}}$ represents the polytope related to the respective formulation $\mathcal{M}$ of the MCI problem.

Let $(V,\mathcal{C})$ be an instance of the MCI problem with $\mathcal{C} = \{V_i \subset V : i \in I = \{1,\dots,n\}\}$. For any $V_i \in \mathcal{C}$ and $S \subset V_i$, we define

$$\delta_i(S) := \delta(\{S, V_i \setminus S\}) = \{\{u,v\} \in E(V_i) : u \in S, v \in V_i \setminus S\}.$$

Moreover, for $i \in I$ and $u \in V_i$, let

$$A_i^+(u) = \{(u,v) \mid v \in V_i \setminus \{u\}\} \quad \text{and} \quad A_i^-(u) = \{(v,u) \mid v \in V_i \setminus \{u\}\}$$

contain all those arcs of $A(V_i)$ which start or end at vertex u, respectively.

Similar to the MST problem, let $\widetilde{\mathcal{T}} \subset \{0,1\}^{|E|}$ denote the set of incidence vectors of all feasible solutions of a given instance $(V,\mathcal{C})$. Because of the assumptions, any feasible solution x has to fulfill the cardinality constraints

$$\sum_{e \in E(V_i)} x_e \geq |V_i| - 1, \qquad i \in I. \tag{4.2}$$

4.1. Flow-based formulation (flow)

The following MILP model of the MCI problem was introduced in [11]. In difference to the MST model, we need to define the flow variables $f^i = (f^i_a)_{a \in A(V_i)}$ and the corresponding constraints for each cluster separately in order to ensure the existence of a spanning tree within the cluster. In particular, for each $i \in I$, a corresponding sink vertex $s_i \in V_i$ has to be defined[2]. Then, the polytope $\widetilde{P}_{flow}$ related to the flow-based formulation of the MCI problem is as follows:

$$\widetilde{P}_{flow} := \{x \in [0,1]^{|E|} : \exists\, f = (f^i)_{i \in I} \text{ with } f^i = (f^i_a)_{a \in A(V_i)} \text{ such that } (x, f) \text{ fulfills (4.2) - (4.5)}\},$$

where

$$\sum_{a \in A_i^-(u)} f^i_a - \sum_{a \in A_i^+(u)} f^i_a = -1, \qquad u \in V_i \setminus \{s_i\},\ i \in I, \tag{4.3}$$

$$f^i_{(u,v)} + f^i_{(v,u)} \leq (|V_i| - 1)x_e, \qquad e = \{u, v\} \in E(V_i),\ i \in I, \tag{4.4}$$

$$f^i_a \geq 0, \qquad a \in A(V_i),\ i \in I. \tag{4.5}$$

This flow-based MILP model possesses a polynomial number of variables and constraints. Similar to the MST problem, the cardinality constraints (4.2) are not necessary, but again, they reduce the set of feasible solutions of the LP relaxation. Moreover, we would like to highlight the fact that the flow propagation condition associated to the sink node $s_i \in V_i$ of cluster V_i, $i \in I$, has already been eliminated as it linearly depends on the remaining conditions of type (4.3).

4.2. Subtour elimination formulation (sub)

In order to adapt the subtour formulation of the MST problem to the MCI problem, we need to regard that (due to possible "interactions" between the clusters) more than $|V_i| - 1$ edges $e \in E(V_i)$ may be contained in a solution. Therefore, we have to introduce extra variables y^i_e, $e \in E(V_i)$, for each cluster $i \in I$. Since P_{sub} has only integral extreme points, the y-variables can be considered to be continuous. Moreover, the interaction of the x- and y-variables has to be modelled. Here, we obtain the polytope

$$\widetilde{P}_{sub} := \{x \in [0,1]^{|E|} : \forall\, i \in I\ \exists\, y^i = (y^i_e)_{e \in E(V_i)} \text{ such that } (x, (y^i)_{i \in I}) \text{ fulfills (4.6) - (4.9)}\},$$

where

$$\sum_{e \in E(V_i)} y^i_e = |V_i| - 1, \qquad i \in I, \tag{4.6}$$

$$\sum_{e \in E(S)} y^i_e \leq |S| - 1, \qquad \emptyset \neq S \subset V_i, i \in I, \tag{4.7}$$

$$y^i_e \leq x_e, \qquad e \in E(V_i), i \in I, \tag{4.8}$$

$$y^i_e \in [0,1] \qquad e \in E(V_i), i \in I. \tag{4.9}$$

[2]Note that the choice of the sink vertices s_i, $i \in I$, can actually influence the strength of the LP bound, see also [11, Sect. 2] for some exemplary calculations. However, as there is no general (dominating) selection rule, here we always choose the lowest-indexed vertex of a cluster, if not stated otherwise.

In this formulation, the number of variables is polynomial, but there are exponentially many constraints. Here, the cardinality constraints (4.2) are not advantageous since they are already induced by the constraints (4.6) and (4.8).

4.3. Cutset formulation (cut)
The adaptation of the cutset formulation of the MST problem to the MCI problem is straightforward: for each cluster, the respective cutset constraints have to be formulated. Hence, the corresponding polytope is as follows:

$$\widetilde{P}_{cut} := \{x \in [0,1]^{|E|} : x \text{ fulfills (4.2) and (4.10)}\},$$

where

$$\sum_{e \in \delta_i(S)} x_e \geq 1, \qquad \emptyset \neq S \subset V_i, i \in I. \tag{4.10}$$

In this formulation only $|E|$ binary variables are used, but an exponential number of constraints (namely $\sum_{i \in I} 2^{|V_i|-1}-1$) is required.

4.4. Multicut formulation (mcut)
The polytope $\widetilde{P}_{mcut}$ related to the multicut formulation of the MCI problem is defined in a similar way:

$$\widetilde{P}_{mcut} := \{x \in [0,1]^{|E|} : x \text{ fulfills (4.2) and (4.11)}\},$$

where

$$\sum_{e \in \delta(\mathcal{P})} x_e \geq |\mathcal{P}| - 1, \qquad \mathcal{P} \in \Pi(V_i),\ i \in I. \tag{4.11}$$

Obviously, also in this formulation, $|E|$ binary variables and an exponential number of constraints are present. Because of the larger set of conditions, it is evident that we have $\widetilde{P}_{mcut} \subseteq \widetilde{P}_{cut}$.

4.5. Martin's formulation (mar)
In order to apply Martin's formulation of the MST problem to the MCI problem we have again to regard that more than $|V_i|-1$ edges of $E(V_i)$ can be contained within the solution. Therefore, additional variables z_e^i, $e \in E(V_i)$, are needed for each cluster $i \in I$. The polytope related to the x-variables is as follows:

$$\widetilde{P}_{mar} := \left\{ x \in [0,1]^{|E|} : \begin{array}{c} \exists\, y = (y_a^i)_{a \in A(V_i)}^{i \in I},\ z = (z_e^i)_{e \in E(V_i)}^{i \in I} \\ \text{such that } (x,y,z) \text{ fulfills (4.12) - (4.17)} \end{array} \right\},$$

where

$$\sum_{e \in E(V_i)} z_e^i = |V_i| - 1, \qquad i \in I, \tag{4.12}$$

$$\sum_{v \in V_i \setminus \{u,w\}} y_{uv,w}^i + z_e^i = 1, \qquad e = \{u,w\} \in E(V_i),\ i \in I, \tag{4.13}$$

$$y_{uv,w}^i + y_{vu,w}^i = z_e^i, \qquad e = \{u,v\} \in E(V_i \setminus \{w\}), w \in V_i, i \in I, \tag{4.14}$$

$$z_e^i \leq x_e, \qquad e \in E(V_i), i \in I, \tag{4.15}$$

$$y_{uv,w}^i \in [0,1], \qquad \{u,v\} \in E(V_i \setminus \{w\}), w \in V_i, i \in I, \tag{4.16}$$

$$z_e^i \in [0,1], \qquad e \in E(V_i), i \in I. \tag{4.17}$$

Since the polytope P_{mar}, related to Martin's formulation of the MST problem, coincides with the convex hull of all spanning trees, the z_e^i-variables are not required to be integer. Moreover, Martin's formulation of the MCI problem yields a MILP model with a polynomial number of variables and constraints.

4.6. Relations between the polytopes

In this subsection, relations between the polytopes associated with the five formulations of the MCI problem are considered. As already mentioned in the previous section, the following inclusions hold for the polytopes of the MST problem:

$$\text{conv}(\mathcal{T}) = P_{mar} = P_{sub} = P_{mcut} \subseteq P_{cut} \subseteq P_{flow}. \tag{4.18}$$

A similar result can be shown for the polytopes defined above for the MCI problem. To this end, let the instance $(V, \mathcal{C})$ of the MCI problem be given. Then, for each cluster $i \in I$ a spanning tree has to be ensured. According to a particular formulation $\mathcal{M} \in \{sub, cut, mcut, flow, mar\}$, any solution x of the MCI problem restricted to the cluster i, has to belong to the respective polytope $P_{\mathcal{M}}(i)$ of the MST problem with vertex set V_i, $i \in I$.

Theorem 6. *For any instance $(V, \mathcal{C})$ of the MCI problem the following holds:*

$$\text{conv}(\widetilde{\mathcal{T}}) \subseteq \widetilde{P}_{sub} = \widetilde{P}_{mar} = \widetilde{P}_{mcut} \subseteq \widetilde{P}_{cut} \subseteq \widetilde{P}_{flow}.$$

Proof. First of all, it is clear by construction that $\text{conv}(\widetilde{\mathcal{T}})$ is contained in every polytope. In order to prove the main part of the statement, we show the following: Let $\mathcal{M}$ and $\mathcal{M}'$ be two formulations of the MST and MCI problem. If $P_{\mathcal{M}}(i) \subseteq P_{\mathcal{M}'}(i)$ holds for all $i \in I$, then we have $\widetilde{P}_{\mathcal{M}} \subseteq \widetilde{P}_{\mathcal{M}'}$.

Since the polytope $P_{\mathcal{M}}(i)$ is contained in a subspace of $[0,1]^{|E|}$ with dimension $|E(V_i)|$, we have to extend $P_{\mathcal{M}}(i)$ to the full space $[0,1]^{|E|}$. To this end, the following multivalued mapping

$$q_i : P_{\mathcal{M}}(i) \longrightarrow [0,1]^{|E|}$$

is considered for all $i \in I$, where

$$q_i(x) := \{x' \in [0,1]^{|E|} : x'_e = x_e, \forall e \in E(V_i)\}.$$

We further define

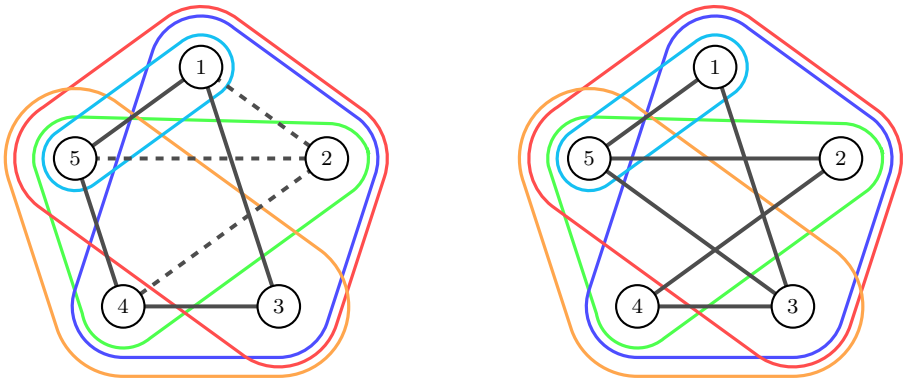
$$\widetilde{P}_{\mathcal{M}}(i) = \bigcup_{x \in P_{\mathcal{M}}(i)} q_i(x), .$$

Hence, $\widetilde{P}_{\mathcal{M}}(i)$ is the extension of polytope $P_{\mathcal{M}}(i)$ to the space $[0,1]^{|E|}$. Consequently, the polytope of the MCI problem related to formulation $\mathcal{M}$ can be described as follows:

$$\widetilde{P}_{\mathcal{M}} = \bigcap_{i \in I} \widetilde{P}_{\mathcal{M}}(i).$$

Since $P_{\mathcal{M}}(i) \subseteq P_{\mathcal{M}'}(i)$ is assumed for every $i \in I$, we have $\widetilde{P}_{\mathcal{M}}(i) \subseteq \widetilde{P}_{\mathcal{M}'}(i)$ for all clusters, and therefore, the inclusion $\widetilde{P}_{\mathcal{M}} \subseteq \widetilde{P}_{\mathcal{M}'}$ holds. Due to the subset relations in (4.18) for the MST polytopes, the statement of the theorem follows. □

Importantly, the relationship of the polytopes of different MST formulations, expressed in (4.18), suggests that an equality exists between $\text{conv}(\mathcal{T})$ and P_{mar}, P_{sub}, and P_{mcut}. However, this equality is changed into inclusion ($\subseteq$) for the corresponding relationship of the MCI problem, described by Theorem 6. Here an example is presented to show that $\text{conv}(\widetilde{\mathcal{T}}) \subset \widetilde{P}_{sub}$ can be true.



(a) A solution to the relaxation of the subtour elimination formulation

(b) An optimal solution of the instance

Figure 2: An illustration of the instance described in Remark 7, where each color refers to one cluster. Here, a continuous line represents an edge with value 1 in the solution, whereas a dotted line shows an edge with value 0.5

Remark 7. *Consider an MCI instance $(V, \mathcal{C})$ with*

$$V := \{1, \dots 5\} \quad \textit{and} \quad \mathcal{C} := \{\{1,2,3,4\}, \{2,4,5\}, \{1,2,3,5\}, \{3,4,5\}, \{1,5\}\}$$

as depicted in Figure 2. By ignoring the integrality restrictions on the x-variables, a solution of the subtour elimination model, in terms of x-variables, can be obtained by

$$x_{13} = x_{15} = x_{34} = x_{45} = 1 \quad \textit{and} \quad x_{12} = x_{25} = x_{24} = 0.5,$$

while all the other x-values are equal to zero (see Figure 2a). The objective value of the linear relaxation of the subtour elimination model is 5.5. On the other hand, any optimal solution has an objective value equal to 6, for instance given by

$$x_{13} = x_{15} = x_{24} = x_{25} = x_{34} = x_{35} = 1,$$

while all the other x-variables are zero (see Figure 2b). Therefore, it can be concluded that $\text{conv}(\widetilde{\mathcal{T}}) \subset \widetilde{P}_{sub}$ holds in this setting.

5. Simple model reduction

As stated above, the (M)ILP models of the MCI problem, proposed in the previous section, possess a considerable number of variables and constraints which, obviously, has consequences for their solution expenses. Therefore, in the following, we aim at finding possibilities to reduce these numbers.

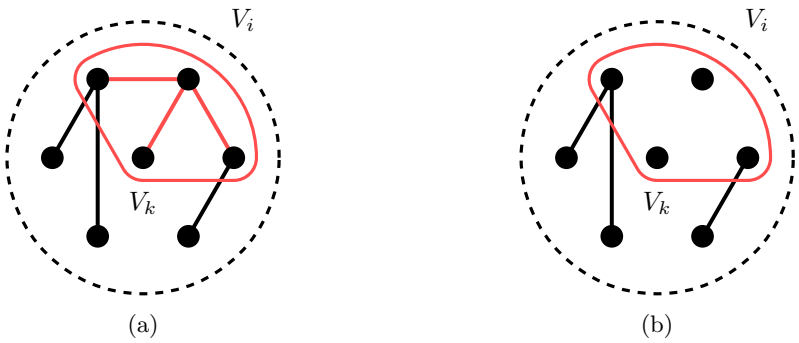


Figure 3: An application of simple model reduction to a cluster V_i of an MCI instance. A cluster V_i (given in the black circle) and its subset cluster V_k (enclosed in a red border) are represented in this figure. In Figure 3a the (ordinary) graph which any MCI formulation induces for cluster V_i is described, whereas Figure 3b represents the (reduced) graph which the same formulation requires for cluster V_i, when the simple model reduction is applied to it. In the latter case, the subset cluster V_k is treated as a single super-vertex.

The main idea is as follows. Consider an instance $(V, \mathcal{C})$ of the MCI problem, and let $E^\star \subseteq E$ denote a corresponding solution. As per definition, the set $E^\star$ leads to the graph $G^\star = (V, E^\star)$ in which each induced subgraph $G^\star[V_i]$ is connected. In particular, if a cluster V_k is a proper subset of another cluster V_i, $(i, k \in I)$, then the graph $G^\star[V_k]$ is a connected component of the graph $G^\star[V_i]$. Therefore, in order to guarantee the spanning tree of cluster V_i, besides the constraints which already ensure the connectivity of V_k, we only have to enforce the connectivity of the vertices in $V_i \setminus V_k$ among each other and with those in V_k. In other words, the respective conditions of cluster V_k can be inherited to formulate all conditions for cluster V_i. This phenomenon can be applied to reduce the overall number of constraints and variables (in case the formulation contains more than only the x-variables) of the models. We refer to this technique as *simple model reduction*, see also Figure 3.

In this section, we discuss the model reduction for all mathematical models of the MCI problem considered above. The performance of the model reductions is also part of our computational experiments (see Section 7). We first introduce the following notations: For a given cluster $V_i \in \mathcal{C}$ let $J_i := \{j \in I : V_j \subset V_i, j \neq i\}$ collect the indices of all proper subset clusters. If $J_i \neq \emptyset$ holds, then let $V_i^{\max}$ be an arbitrary element of $\{V_k : k \in J_i, |V_k| \geq |V_j| \, \forall j \in J_i\}$, otherwise we set $V_i^{\max} = \emptyset$. Furthermore, we define the following:

$$V_i^R = \begin{cases} V_i \setminus V_i^{\max} \cup \{v_0\}, & \text{if } J_i \neq \emptyset, \\ V_i, & \text{otherwise,} \end{cases}$$

where v_0 is introduced to represent an artificial vertex, hereinafter referred to as a *super-vertex*. To keep the notation simple, we do not attach a further index $i \in I$ to a super-vertex v_0. Later, in the models, it is clear from the context which cluster (and super-vertex) is currently considered.

For the sake of simplicity, in the following we again renounce adding the objective function $e^\top x \to \min$ and the integer condition $x_e \in \{0, 1\}$, $e \in E$, to each model of the MCI problem. To avoid that an instance $(V, \mathcal{C})$ decomposes into a number of simpler MCI instances, we

assume that

$$I_1 \cap I_2 = \emptyset \ \wedge \ I_1 \cup I_2 = I \quad \Rightarrow \quad \left| \bigcup_{i \in I_1} V_i \cap \bigcup_{i \in I_2} V_i \right| \geq 2$$

holds for any $I_1, I_2 \subset I$.

5.1. Flow-based formulation

The simple model reduction can be applied to the flow-based model in two ways. Here we present both the methods of application.

5.1.1. The first variant

In the first variant of a reduced flow-based model constraints (4.3) - (4.5) are replaced by the following conditions:

$$\sum_{a \in A_i^-(u)} f_a^i - \sum_{a \in A_i^+(u)} f_a^i = -1, \qquad u \in V_i \setminus V_i^{\max},\, u \neq s_i, i \in I, \tag{5.1}$$

$$f_{(u,v)}^i + f_{(v,u)}^i \leq (|V_i^R| - 1)x_e, \qquad e = \{u,v\} \in E(V_i) \setminus E(V_i^{\max}), i \in I, \tag{5.2}$$

$$f_a^i \geq 0, \qquad a \in A(V_i) \setminus A(V_i^{\max}), i \in I. \tag{5.3}$$

Note that we use $s_i = v_0$ if $V_i^{\max} \neq \emptyset$ holds, whereas for $V_i^{\max} = \emptyset$ the lowest-indexed vertex is chosen as the sink. Besides, for each $i \in I$ with $J_i \neq \emptyset$, about $|V_i^{\max}|^2$ f_a^i-variables and related constraints could be saved.

5.1.2. The second variant

The second variant of the reduced flow-based formulation uses even fewer f-variables for each cluster. To this end, we define the following abbreviation:

$$\tilde{A}_i = \{(u,v) : u, v \in V_i^R, u \neq v\}.$$

Then, constraints (4.3) - (4.5) can be replaced by:

$$\sum_{a \in \tilde{A}_i^-(u)} f_a^i - \sum_{a \in \tilde{A}_i^+(u)} f_a^i = -1, \quad u \in V_i \setminus V_i^{\max},\, u \neq s_i, i \in I, \tag{5.4}$$

$$f_{(u,v)}^i + f_{(v,u)}^i \leq \begin{cases} (|V_i^R| - 1) \sum\limits_{l \in V_i^{\max}} x_{\{u,l\}} & \text{if } u \neq v_0, v = v_0, \\ (|V_i^R| - 1)x_{\{u,v\}} & \text{if } u \neq v_0, v \neq v_0, \end{cases} \qquad \{u,v\} \in E(V_i^R), i \in I, \tag{5.5}$$

$$f_a^i \geq 0, \quad a \in \tilde{A}_i, i \in I. \tag{5.6}$$

Here the set $V_i^{\max}$ is directly replaced by the super-vertex v_0. Moreover, if $J_i = \emptyset$ holds, then s_i is chosen as described above, and v_0 is considered as a vertex not belonging to V_i.

Remark 8. *Note that preliminary computational tests have shown that the second variant of the flow-based formulation (simply called F2) performs better than the first version (F1), see Table 1 for some representative results. In this table each row is the average of* 10 *instances, belonging to different instance types (which are fully specified later in Section 7),*

and the column $|E^*|_\varnothing$ represents the average optimal value. Remarkably, the advantages of the second version (F2) can be observed in almost all the reported aspects, i.e., the number of constraints, the number of variables, and the runtime. As regards the LP bound, both versions show equivalent results. Hence, the second version possesses a less complex structure, while the quality of the LP bound is maintained. Based on these observations, only the second variant is later used in this article for the numerical comparison of the different formulations, where it is then simply denoted by F (instead of F2).

Table 1: Comparison (based on preliminary tests) of two variants of the flow-based formulation for simple model reduction

				Constraints		Variables		LP relaxation		Runtime (sec)	
Type	$\|V\|$	$\|\mathcal{C}\|$	$\|E^*\|_\varnothing$	F1	F2	F1	F2	F1	F2	F1	F2
1	13	13	15.40	315.20	229.40	441.50	355.70	14.46	14.46	0.26	0.22
		65	26.50	1197.40	674.10	1260.00	736.70	24.65	24.65	1.59	1.07
	15	15	18.40	502.90	362.00	715.20	574.30	17.31	17.31	0.42	0.31
		75	30.90	1978.70	1088.60	2206.30	1316.20	28.30	28.30	3.70	3.20
	17	17	21.40	722.60	536.20	1063.80	877.40	19.81	19.81	1.31	0.94
		85	38.60	2753.90	1570.60	3215.00	2031.70	35.05	35.05	8.71	6.21
3	13	13	14.70	364.90	265.30	503.10	403.50	13.96	13.96	0.26	0.22
		65	23.90	1384.50	876.60	1532.60	1024.70	21.96	21.96	3.39	2.54
	15	15	17.00	587.40	439.30	844.00	695.90	16.10	16.10	0.62	0.49
		75	29.80	2100.70	1276.50	2412.70	1588.50	27.21	27.21	6.70	6.04
	17	17	20.90	837.70	657.90	1247.20	1067.40	19.39	19.39	2.32	1.57
		85	32.50	3430.80	2213.60	4248.40	3031.20	29.16	29.16	85.62	75.66

5.2. Subtour elimination formulation

Applying the idea of model reduction, the reduced model of the subtour formulation is as follows:

$$\sum_{e\in E(S)} y_e^i \le |S| - 1, \qquad \emptyset \neq S \subset V_i^R, i \in I, \tag{5.7}$$

$$\sum_{e\in E(V_i^R)} y_e^i = |V_i^R| - 1, \qquad i \in I, \tag{5.8}$$

$$y_{\{u,v\}}^i \le \begin{cases} \sum\limits_{k\in V_i^{\max}} x_{\{k,v\}} & \text{if } u = v_0, v \neq v_0, \\ x_{\{u,v\}} & \text{if } u \neq v_0, v \neq v_0, \end{cases} \qquad \{u,v\} \in E(V_i^R), i \in I, \tag{5.9}$$

$$y_e^i \in [0,1], \qquad e \in E(V_i^R), i \in I. \tag{5.10}$$

Since, in case of $J_i \neq \emptyset$, v_0 represents a set of vertices, we obtain the particular form of the conditions in (5.9), which is an alternative to (4.8). Indeed, several edges can be incident with vertices in $V_i^{\max}$.

5.3. Cutset formulation

In order to apply model reduction to the cutset formulation we need another notation to describe the appropriate edge set related to V_i^R, $i \in I$:

$$\widetilde{\delta}_i(S) := \begin{cases} (S \setminus \{v_0\} \cup V_i^{\max}) \times (V_i^R \setminus S), & \text{if } v_0 \in S, \\ S \times V_i \setminus S, & \text{otherwise,} \end{cases} \quad S \subset V_i^R.$$

Then, the reduced set of constraints of the cutset formulation is the following:

$$\sum_{e \in \widetilde{\delta}_i(S)} x_e \geq 1, \qquad \emptyset \neq S \subset V_i^R, i \in I. \tag{5.11}$$

Notably, the number of variables for the cutset formulation is not reduced by simple model reduction.

5.4. Multicut formulation

Similar to above, we need to introduce an appropriate notation:

$$\widetilde{\Pi}(V_i) = \{\mathcal{P} \in \Pi(V_i) : \text{ if } J_i \neq \emptyset \text{ then } V_i^{\max} \subseteq \rho_s \text{ for some } \rho_s \in \mathcal{P}\}.$$

Then, the reduced multicut formulation of the MCI problem possesses the following restrictions:

$$\sum_{e \in \delta(\mathcal{P})} x_e \geq |\mathcal{P}| - 1, \qquad \mathcal{P} \in \widetilde{\Pi}(V_i),\ i \in I. \tag{5.12}$$

Obviously, also in this formulation, there is no decrease in the number of variables.

5.5. Martin's formulation

Using v_0 instead of $V_i^{\max}$ (if $J_i \neq \emptyset$ holds) leads to the following constraints of the reduced version of Martin's MILP model:

$$\sum_{\{u,v\} \in E(V_i^R)} z^i_{\{u,v\}} = |V_i^R| - 1, \quad i \in I, \tag{5.13}$$

$$\sum_{v \in V_i^R \setminus \{u,w\}} y^i_{uv,w} + z^i_{\{u,w\}} = 1, \quad u, w \in V_i^R, u \neq w,\ i \in I, \tag{5.14}$$

$$y^i_{uv,w} + y^i_{vu,w} = z^i_{\{u,v\}}, \quad \{u,v\} \in E(V_i^R \setminus \{w\}), w \in V_i^R, i \in I, \tag{5.15}$$

$$z^i_{\{u,v\}} \leq \begin{cases} \sum\limits_{k \in V_i^{\max}} x_{\{k,v\}} & \text{if } u = v_0, v \neq v_0, \\ x_{\{u,v\}} & \text{if } u \neq v_0, v \neq v_0, \end{cases} \quad \{u,v\} \in E(V_i^R), i \in I, \tag{5.16}$$

$$y^i_{uv,w} \in [0,1], \quad \{u,v\} \in E(V_i^R \setminus \{w\}), w \in V_i^R, i \in I, \tag{5.17}$$

$$z^i_{\{u,v\}} \in [0,1], \quad \{u,v\} \in E(V_i^R), i \in I, \tag{5.18}$$

Note that the replacement of $V_i^{\max}$ by v_0 (if $J_i \neq \emptyset$ is true) requires the special form of conditions (5.16) since several edges in the solution can be incident with vertices in $V_i^{\max}$.

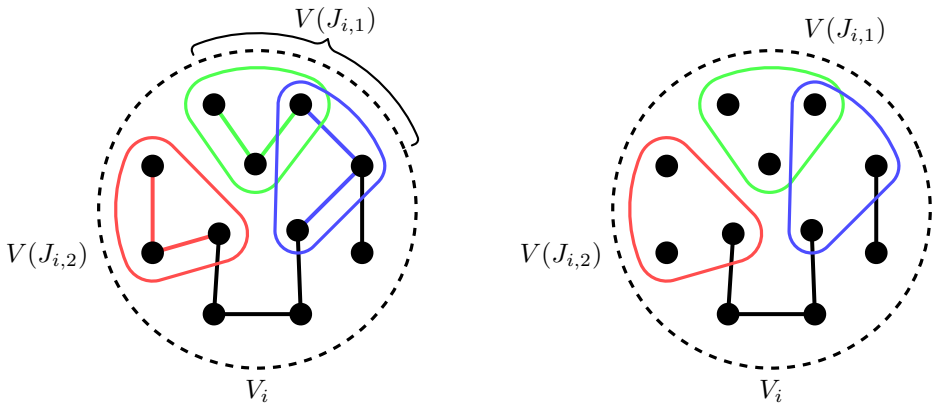


Figure 4: The benefit of applying the generalized model reduction to a formulation of the MCI problem is illustrated in this figure. For a given instance of the MCI problem the figure shows a cluster V_i with $J_i \neq \emptyset$, where the graph $\mathcal{G}[J_i]$ has two connected components. One component is labelled as $V(J_{i,1})$ (consisting of a green and a blue cluster), whereas the other one is named as $V(J_{i,2})$ (consisting of a single cluster in red colour). Figure 4a represents the graph which a given MCI formulation ensures when we apply no model reduction. Likewise, 4b depicts the effect of applying the generalized model reduction leading to several super-vertices.

6. Generalized model reduction

In the previous section, a single cluster, termed $V_i^{\max}$, is used for cluster V_i (if existing) to save various constraints and possibly several variables. This idea can be applied in a generalized form if a certain cluster contains more than one proper subset cluster.

Let $(V, \mathcal{C})$ be an instance of the MCI problem, which is defined on the graph $G = (V, E)$. For a cluster $i \in I$ with $J_i \neq \emptyset$, let $\mathcal{G}[J_i]$ denote an induced graph defined as follows: $J_i \subset I$ is the vertex set and $\{j, k\}$ $(j \neq k \in J_i)$ belongs to the edge set if and only if $V_j \cap V_k \neq \emptyset$. We denote the connected components of $\mathcal{G}[J_i]$ by the index sets $J_{i,k} \subset J_i$, $k = 1, \dots \gamma_i$, where γ_i is the number of connected components. Moreover, if $E^\star \subseteq E$ is a solution of the instance $(V, \mathcal{C})$, then it leads to the graph $G^\star = (V, E^\star)$, that enables each induced graph $G^\star[V_i]$ to be connected for all $i \in I$. Furthermore, it is easy to see that, if $|J_{i,k}| > 1$ holds for some k, then the connectivity of $G^\star[V(J_{i,k})]$ is guaranteed by the induced graphs $G^\star[V_j]$, $j \in J_{i,k}$ (see [11, Lemma 3.1]). Hence, similar to the previous section, the vertex set of each connected component $G^\star[V(J_{i,k})]$, $k = 1, \dots, \gamma_i$, can be considered as a super-vertex. This form of model reduction (considering each connected component as a super-vertex) is called the *generalized model reduction*, see Figure 4 fore a more detailed illustration.

Note that already in [11] connected components of $\mathcal{G}[J_i]$ for a cluster V_i are considered to define some *instance reduction rules*. There, in particular the case $V(J_i) = V_i$ is exploited. For every cluster $V_i \in \mathcal{C}$, let

$$V_i^\star := \begin{cases} \{v_1^\star, \dots, v_{\gamma_i}^\star\}, & \text{if } J_i \neq \emptyset, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Then, the notation V_i^R of the previous section is redefined as $V_i^R = V_i \backslash V(J_i) \cup V_i^\star$. Although for all formulations the same principle of generalized model reduction is applied, we present the resulting constraints in detail to better see the effects and savings.

6.1. Flow-based formulation

Applying the idea of generalized model reduction to the flow-based formulation leads to the following constraints:

$$\sum_{a \in \tilde{A}_i^-(u)} f_a^i - \sum_{a \in \tilde{A}_i^+(u)} f_a^i = -1, \quad u \in V_i^R,\, u \neq s_i, i \in I, \tag{6.1}$$

$$f_{(u,v)}^i + f_{(v,u)}^i \leq \begin{cases} (|V_i^R| - 1) \sum\limits_{\{p,q\} \in V(J_{i,k}) \times V(J_{i,s})} x_{\{p,q\}}, & \text{if } u = v_k^\star \in V_i^\star, v = v_s^\star \in V_i^\star, \\ (|V_i^R| - 1) \sum\limits_{p \in V(J_{i,k})} x_{\{u,p\}}, & \text{if } u \notin V_i^\star, v = v_k^\star \in V_i^\star, \\ (|V_i^R| - 1) x_{\{u,v\}}, & \text{if } u \notin V_i^\star, v \notin V_i^\star, \end{cases} \tag{6.2}$$

$$\{u,v\} \in E(V_i^R), i \in I,$$

$$f_a^i \geq 0, \quad a \in \tilde{A}_i, i \in I, \tag{6.3}$$

We set $s_i = v_1^\star \in V_i^\star$ if $J_i \neq \emptyset$ holds. The set of constraints (6.2) is a variant of constraints (4.4). However, to write (6.2) we have to consider three cases related to edge $\{u,v\} \in E(V_i^R)$ that are based upon possible ways in which the vertices of edge $\{u,v\}$ can include the super-vertices. In general, estimating the savings of variables and constraints is difficult for generalized model reduction. However, if a subset $\widetilde{V} \subseteq V_i$ exists which is the union of smaller clusters, then (in principle) all variables and constraints of V_i belonging to $\widetilde{V}$ can be saved. Some average information will be given in the computational part of this paper.

6.2. Subtour elimination formulation

The formulation of the related reduced model requires several changes and results to:

$$\sum_{e \in E(V_i^R)} y_e^i = |V_i^R| - 1, i \in I, \tag{6.4}$$

$$\sum_{e \in E(S)} y_e^i \leq |S| - 1, \emptyset \neq S \subset V_i^R, i \in I, \tag{6.5}$$

$$y_{\{u,v\}}^i \leq \begin{cases} \sum\limits_{\{p,q\} \in V(J_{i,k}) \times V(J_{i,s})} x_{\{p,q\}} & \text{if } u = v_k^\star \in V_i^\star, v = v_s^\star \in V_i^\star, \\ \sum\limits_{p \in V(J_{i,k})} x_{\{u,p\}} & \text{if } u \notin V_i^\star, v = v_k^\star \in V_i^\star, \\ x_{\{u,v\}} & \text{if } u \notin V_i^\star, v \notin V_i^\star, \end{cases} \quad \begin{array}{l} \{u,v\} \in E(V_i^R), \\ i \in I, \end{array} \tag{6.6}$$

$$y_e^i \in [0,1],\ e \in E(V_i^R),\ i \in I. \tag{6.7}$$

Clearly, also here we have to differentiate between regular vertices or super-vertices as can be seen by three different cases of (6.6). The application of the subtour formulation of the MST problem to the MCI problem required the introduction of a lot of y-variables. However, similar to the previous subsection, the usage of the generalized model reduction possibly can rapidly reduce their number and that of the constraints (see Section 7).

6.3. Cutset formulation

In order to present the respective constraints for the cutset formulation with generalized model reduction, a redefinition of $\widetilde{\delta}_i$, already used in Section 5, is required. To this end, for $S \subset V_i^R$, $i \in I$, we define $\widetilde{S}$ as follows:

$$\widetilde{S} = \left(\bigcup_{k \in S \cap V_i^\star} V(J_{i,k}) \right) \cup (S \setminus V_i^\star).$$

Then, we have the following new definition of $\widetilde{\delta}_i$:

$$\widetilde{\delta}_i = \left\{ \{u,v\} \in E(V_i) : u \in \widetilde{S}, v \in V_i \setminus \widetilde{S} \right\}.$$

As a consequence, the cutset formulation of this section remains the same as in the previous section. The only change is the new definition of $\widetilde{\delta}_i$. Therefore, we omit a repetition of the formulas.

6.4. Multicut formulation

As done earlier with the other notations, the definition of $\widetilde{\Pi}(V_i)$ also requires a refinement:

$$\widetilde{\Pi}(V_i) = \{\mathcal{P} \in \Pi(V_i) : \nexists k \in \{1,\dots,\gamma_i\} \text{ s.t. } V(J_{i,k}) \cap \rho_s \neq \emptyset \wedge V(J_{i,k}) \cap \rho_t \neq \emptyset, \forall \rho_s \neq \rho_t \in \mathcal{P}\},$$

i.e., there is no partition such that two of its subsets ρ_s and ρ_t have a non-empty intersection with one of the (vertex sets of the γ_i) components of $\mathcal{G}[V_i]$. Altogether, the variant of the multicut formulation for the current section also remains the same as that of Section 5, but with the new definition of $\widetilde{\Pi}(V_i)$.

6.5. Martin's formulation

The constraints of the respective formulation of Martin's model are the following:

$$\sum_{\{u,v\} \in E(V_i^R)} z^i_{\{u,v\}} = |V_i^R| - 1, \quad i \in I, \tag{6.8}$$

$$\sum_{v \in V_i^R \setminus \{u,w\}} y^i_{uv,w} + z^i_{\{u,w\}} = 1, \quad u,w \in V_i^R, u \neq w,\ i \in I, \tag{6.9}$$

$$y^i_{uv,w} + y^i_{vu,w} = z^i_{\{u,v\}}, \quad \{u,v\} \in E(V_i^R \setminus \{w\}), w \in V_i^R, i \in I, \tag{6.10}$$

$$z^i_{\{u,v\}} \leq \begin{cases} \sum\limits_{\{p,q\} \in V(J_{i,k}) \times V(J_{i,s})} x_{\{p,q\}} & \text{if } u = v_k^\star \in V_i^\star, v = v_s^\star \in V_i^\star, \\ \sum\limits_{p \in V(J_{i,k})} x_{\{u,p\}} & \text{if } u \notin V_i^\star, v = v_k^\star \in V_i^\star, \\ x_{\{u,v\}} & \text{if } u \notin V_i^\star, v \notin V_i^\star, \end{cases} \tag{6.11}$$

$$\{u,v\} \in E(V_i^R), i \in I,$$

$$y^i_{uv,w} \in [0,1], \quad \{u,v\} \in E(V_i^R \setminus \{w\}), w \in V_i^R, i \in I, \tag{6.12}$$

$$z^i_{\{u,v\}} \in [0,1], \quad \{u,v\} \in E(V_i^R), i \in I, \tag{6.13}$$

In conditions (6.11) again three possibilities, corresponding to the vertices of edge $\{u,v\} \in E(V_i^R)$, have to be regarded. This is similar to the flow-based and subtour formulation of this section.

7. Computational analysis

This section contains a detailed computational analysis of the different MCI models and the effectiveness of the model reduction principles. After the description of the experimental environment in Subsection 7.1, we explain the aim of our computational comparisons and give the respective notations in Subsection 7.2. The detailed analysis is then presented in the last subsection.

7.1. Experimental environment

In order to compare the different models of the MCI problem and to investigate the impact of the proposed model reductions, we consider two kinds of instances, termed *kind 1* and *kind 2*. Instances of *kind 1* have at least as many clusters as vertices, whereas those of *kind 2* have fewer clusters than vertices.

In [11] *kind 1* instances are already used to analyze the performance of flow-based models of the MCI problem together with the effects of *instance reduction rules*. Therein, four types of instances are considered in which the cardinality of each cluster $V_i \in \mathcal{C} = \{V_1, \dots, V_n\}$ is randomly chosen according to a uniform distribution in the following sense:

$$\begin{array}{ll} \text{Type 1:} & |V_i| \in \{2, \dots, m\}, \\ \text{Type 2:} & |V_i| \in \{2, \dots, \lfloor m/2 \rfloor\}, \\ \text{Type 3:} & |V_i| \in \{\lceil m/4 \rceil, \dots, m\}, \\ \text{Type 4:} & |V_i| \in \{\lceil m/4 \rceil, \dots, \lfloor m/2 \rfloor\}. \end{array} \tag{7.1}$$

Then, for given values $m = |V|$ and $n = |\mathcal{C}|$ the cluster V_i is chosen as a random subset of $V = \{1, \dots, m\}$. Note that instances of types 1 and 2 can have very small clusters (in contrast to types 3 and 4), while types 1 and 3 possibly contain very large clusters (compared to types 2 and 4).

To generate instances of *kind 2* three parameters are used. For a desired number n of clusters the number m of vertices is defined by $m := p \cdot n$ with $p \in \{2, 3, \dots\}$. Moreover, the maximum vertex frequency (the number of clusters containing the vertex) is controlled by the parameter $k \in \{3, \dots, 6\}$. The detailed procedure to generate *kind 2* instances is given in Appendix A.

To solve an MCI instance by a given formulation, first the preprocessing, including the application of instance and model reductions is done in MATLAB® Release R2016a. This generates the (M)ILP model of the given instance corresponding to the respective formulation. Later on, from MATLAB® the CPLEX® (Version 12.6.3) routine `cplexmilp` is called to solve the (M)ILP model. Moreover, for preprocessing, the MATLAB® built-in feature of sparsity is applied to reduce the required memory for generating the (M)ILP model. The computer used is equipped with an Intel® Xeon® processor X5670 at 2.93 GHz with 96 GB of memory. Furthermore, a time limit of 3600 seconds is used for the solution of an instance.

7.2. Methodology and notations

The aim of our computational tests is twofold. On the one hand, we want to compare the performance of the presented different MCI formulations, mainly by means of the number of instances solved to optimality and the computational time needed to solve the instances. On the other hand, we analyze the effects of the two model reduction principles. For this

purpose, we consider the *flow, cutset, subtour,* and *Martin's* formulation. Due to the huge number of constraints of the *multi-cut* formulation (in comparison to the other approaches), we do not perform corresponding calculations. Because of the benefits of instance reduction rules reported in [10, 11] we also include them with different combinations of model reduction rules. To get a better overview on the structural properties of the considered approaches, some of the tables also contain additional information about the numbers of variables and constraints (for the different stages of reduction), as well as the LP values.

Note that, in Tables 2-4, instances of *kind 1* are used for the tests, whereas in Table 5 those of *kind 2* are taken. Moreover, Table 6 is based on a representative selection of instances from both categories, i.e., *kind 1* and *kind 2*.

To provide a complete overview, all of the notations used in the subsequent tables are already introduced here; but, to improve the comprehension, some of them are later repeated again close to the place where they appear first:

- The heading **Type** indicates the type of the instance, described in (7.1).
- The column $|\boldsymbol{E}^\star|_\varnothing$ contains the average optimal values of the instances solved within the time limit of 3600 seconds (averages over 10 random instances, in general).
- The multi-column **Runtime** shows the average runtimes needed to solve an instance after the application of the respective reduction technique(s). The columns **NR**, **sMR**, **IR**, **IsMR**, and **IgMR** contain the average runtime, if either no reduction rule, only simple model reduction, only instance reduction, instance reduction followed by simple model reduction, or instance reduction followed by generalized model reduction is applied, respectively.
- The columns **F**, **C**, **S**, and **M** represent the flow, cutset, subtour, and Martin's formulations, respectively.
- The mult- column **%age red IR** represents the percentages of reduction of the number of vertices (column **vert**) and clusters (column **clus**), by instance reduction. Similarly, in Table 6, the multi-column **%age red IsMR** and **%age red IgMR** describe the percentages of reduction of the constraints (multi-column **cons**) and variables (multi-column **var**), by the two combinations IsMR and IgMR, both measured with respect to a level of 100% defined by the status after instance reductions are performed.
- If out of 10 instances not all could be solved[3] within the time limit of 3600 seconds, then below the average runtime the number of successfully solved instances is expressed as a **bracketed number** [x]. In such a case the average runtime is found over these solved instances (whose total number is indicated within the bracket).
- According to the generation of instances of *kind 2*, the number of clusters can be smaller than desired. Therefore, additionally, the average number of clusters is given in Table 5 in the column with heading $|\mathcal{C}|_\varnothing$.

In Table 6, the additional abbreviation **IR (#)** stands for the number of constraints or variables possessed by a given model after the application of instance reductions. Furthermore, in each table dealing with runtimes, we will use boldface to highlight the best formulation. By *best*, we mean that this specific approach could solve the largest number of instances to optimality, where ties are broken by choosing the smallest average computation time.

[3]Note that this usually refers to instances, where either no feasible point is found or the optimality (of a feasible solution) could be proved within the time limit. On the other hand, in Table 5, for the row corresponding to $m = 50$, $n = 10$, $p = 5$, and $k = 5$, we also observed that instances (i.e., the related branching trees) can become too complex leading to the CPLEX message 'out of memory'.

7.3. Results and discussions

In order to get a first overview on the computational behavior of the proposed reductions, Table 2 corresponds to *kind 1* instances with $m = 15$ vertices and up to 75 clusters. For these sets of instances, we briefly compare the formulations presented in Sections 4 and 5, in terms of their running times and the numbers of instances solved to optimality, for five different stages of reduction. More precisely, as a short repetition of the previous section, we perform

- no reduction at all (columns 5-8),
- only simple model reduction (columns 9-12),
- only instance reduction from [11] (columns 13-16),
- instance reduction together with simple model reduction (columns 17-20),
- instance reduction together with generalized model reduction (columns 21-24).

In particular, we would like to highlight the following key observations:

- In a non-reduced setting (columns 5-8), the flow-based approach is the only model that is able to cope with all instances within the given time limit. Moreover, in almost all cases, this formulation provides the best solution times, whereas the other approaches (C, S, and M) struggle to solve some instances of type 1 and type 3. Note that these types allow for instances having very large clusters, so that the non-reduced versions (of C, S, and M) are likely to form rather complex (M)ILPs (having exponentially many or at least a higher polynomial number of constraints) to be solved. Contrary to that, the manageable polynomial number of variables and constraints in the flow-based formulation actually counterbalances the fact that it typically possesses the worst LP bound among all the tested approaches. As regards the instances of type 2 and type 4, there are only slight differences between the various solution times.
- After applying simple model reductions (columns 9-12), also the previously harder instance categories (type 1 and type 3) can be tackled by all the formulations in reasonably small runtimes. Note that, for these instance types, the potentials of successful model reduction are particularly high since we also have larger clusters, so that the required subset relations between the given clusters (to perform simple model reduction) are more likely to be present. However, we also have to state that the application of simple model reduction leads to additional computational efforts which can sometimes entail a slightly higher runtime compared to the non-reduced setting. In these calculations, this phenomenon could only be noticed in a few cases where the original solution times were already very small. Altogether, the flow-based approach is still very good for almost all cases, while the subtour formulation seems to be favorable for type 2 and the cutset model performs well for type 4, respectively.
- On average, the application of instance reductions (columns 13-16) from [11] is also very helpful compared to the non-reduced setting (columns 5-8). In particular, these reductions mostly decrease the number of clusters and partly that of the vertices (columns 2-3), so that they also contribute to the fact that any instance can now be solved by any model within the time limit. As already pointed out in [10, 11], these reductions can usually be performed in less than one second, so that they should

Table 2: Comparison of the runtime (in seconds) of all four formulations (flow, cutset, subtour and Martin's formulation), for instances of the MCI problem with $|V| = 15$ (for each row 10 instances are considered)

	%age red IR			Runtime (in seconds)																			
				NR				sMR				IR				IsMR				IgMR			
$\|\mathcal{C}\|$	clus	vert	$\|E^*\|_\varnothing$	F	C	S	M	F	C	S	M	F	C	S	M	F	C	S	M	F	C	S	M
Type 1																							
15	9.3	0.0	18.4	**0.5**	264.9	35.5	8.2	**0.8**	2.4	2.5	1.9	**1.3**	92.6	16.1	5.4	**0.4**	2.3	2.5	1.8	**0.3**	2.1	2.0	1.5
30	17.0	0.0	23.5	**3.3**	36.7	286.5	39.1	**1.4**	2.4	1.9	2.3	**2.1**	13.5	32.1	13.8	**0.9**	2.2	1.7	1.9	**0.7**	1.4	1.1	1.4
45	23.8	0.0	27.2	**1.9**	73.3	409.8	77.1	**1.4**	2.3	1.8	2.3	**2.7**	13.1	12.9	7.7	**0.8**	1.9	1.6	1.8	**0.6**	1.0	1.0	1.2
60	27.5	0.0	29.7	**4.7**	78.5	1295 [9]	309.0	**2.6**	2.7	2.4	3.8	**3.5**	9.4	19.5	21.4	**1.5**	2.3	2.2	3.11	1.2	1.3	**1.1**	1.8
75	32.1	0.0	30.9	**6.2**	68.6	756.2	434.2 [9]	2.6	3.1	**2.5**	4.2	**3.1**	8.6	37.9	52.8	**1.6**	2.1	1.8	2.9	**0.9**	1.0	**0.9**	1.4
Type 2																							
15	0.7	2.0	22.7	**0.2**	0.5	**0.2**	0.5	**0.2**	**0.2**	**0.2**	0.5	1.1	1.5	**0.2**	0.5	**0.2**	**0.2**	**0.2**	0.3	**0.2**	**0.2**	**0.2**	0.4
30	0.7	0.0	29.5	**0.4**	0.8	0.6	1.0	0.4	0.4	**0.3**	0.7	2.2	2.6	**0.7**	1.6	0.4	0.4	**0.3**	0.7	**0.3**	0.4	**0.3**	0.6
45	2.2	0.0	35.6	**0.7**	**0.7**	0.8	1.4	0.5	0.5	**0.4**	0.9	**1.5**	2.2	**1.5**	2.1	0.5	0.5	**0.4**	0.8	0.5	0.4	**0.3**	0.6
60	5.3	0.0	39.6	1.3	**0.9**	1.2	2.0	1.0	0.7	**0.6**	1.1	2.3	1.9	**1.8**	2.4	1.0	0.7	**0.6**	1.2	0.9	0.5	**0.4**	0.8
75	8.0	0.0	42.5	1.2	**1.1**	1.4	2.8	1.2	0.9	**0.7**	1.4	1.3	**1.1**	1.3	2.6	1.3	0.8	**0.6**	1.2	0.7	0.5	**0.4**	0.7
Type 3																							
15	24.0	10.0	17.0	**1.1**	703.1	183.3	11.6	**0.4**	3.3	5.0	3.0	**0.9**	193.8	25.9	7.7	**0.5**	3.3	4.7	2.3	**0.3**	3.1	4.5	2.1
30	22.3	0.0	22.0	**4.0**	1752 [8]	646.1	99.5	**2.6**	**2.6**	3.6	4.6	**3.3**	291.1	27.9	19.5	**1.7**	2.6	3.9	4.2	**1.4**	2.4	3.4	3.7
45	24.0	0.0	25.7	**3.8**	1679	899.8	235.5	**1.9**	3.2	4.4	6.2	**3.3**	143.3	39.6	32.1	**1.5**	2.9	4.2	5.4	**1.3**	2.3	3.9	4.3
60	23.2	0.0	28.0	**9.8**	2348 [2]	1509 [5]	1307	5.5	**4.4**	7.7	15.4	**8.9**	1027	96.9	144.6	4.0	**3.8**	7.6	13.6	3.7	**3.6**	6.2	11.6
75	28.7	0.0	29.8	**15.5**	1388 [2]	2696 [1]	2189 [8]	6.2	**4.6**	10.6	21.8	**9.6**	530.5	81.0	168.6	4.2	**3.9**	9.1	15.3	3.9	**3.4**	8.1	12.8
Type 4																							
15	0.0	2.0	21.7	**0.3**	0.7	0.4	0.7	0.8	**0.4**	0.7	0.7	1.6	2.1	**0.3**	0.6	**0.3**	**0.3**	0.4	0.7	0.4	**0.3**	0.4	0.6
30	0.0	0.0	28.7	**0.6**	2.5	0.9	1.6	1.3	**1.1**	1.2	1.9	**1.2**	3.8	1.5	2.3	0.7	**0.6**	0.9	1.6	**0.6**	**0.6**	0.9	1.5
45	0.7	0.0	32.4	2.5	5.9	**1.8**	4.2	3.4	**1.7**	2.3	4.0	4.2	9.2	**2.2**	5.7	2.2	**1.1**	1.7	3.4	2.4	**1.2**	1.6	3.6
60	1.5	0.0	35.7	3.6	10.9	**3.0**	8.2	4.6	**2.5**	3.9	7.9	5.1	15.4	**4.1**	9.6	3.2	**1.9**	3.7	7.0	3.3	**1.8**	3.2	6.4
75	1.6	0.0	38.4	6.0	25.4	**5.4**	19.8	11.2	**3.7**	6.8	14.5	8.0	36.4	**7.3**	22.8	5.1	**3.6**	8.2	15.7	6.6	**2.8**	7.8	13.6

always be taken into account. However, similar to the previous point, here we also notice some few cases where the total runtimes slightly increase, but still remain on a very low level. Based on these observations and the detailed computations in [10], in all the following tables we will at least apply this type of reduction.

- In the vast majority of the cases, the application of instance reductions together with model reductions (columns 17-24) additionally boosts the performances that have been obtained if only one of these reduction paradigms is used. Moreover, although requiring higher efforts for precalculations, the generalized model reduction slightly outperforms the scenario where only simple model reductions are applied. Also here, we can observe that the flow-based models performs very well for all instance types, while the subtour and the cutset formulation are particularly suitable for type 2 and type 4, respectively. For these sets of instances, Martin's formulation never provides the fastest solution times.

Hence, we recommend to use all reduction principles (i.e., instance and generalized model reduction) as this (almost) always leads to the best results in terms of runtimes.

A more detailed look at these exemplary results suggests that, after having performed generalized model reductions, instance types 3 and 4 (having no very small clusters) can be termed to be more challenging than types 1 and 2. Consequently, for the consideration of larger instances (i.e., instances with more than 15 vertices), we split the four types into two subsets each of which being studied in a single table. More precisely, in Table 3, instances of types 1 and 2 having up to 22 vertices are considered, whereas Table 4 shows the results for types 3 and 4, but only up to 18 vertices. In both cases, all available reductions have been performed before trying to solve the instance. Moreover, we also report about the average values of the integer problem and the various LP relaxations. It is worth noting that:

- In both tables, the experimental data support our theoretical observation (given in Theorem 6) about the strength of the LP relaxations. Furthermore, we see that the relative gaps (i.e., the percentaged differences between the true optimal value and the LP bound) are rather small in all cases, usually not exceeding (roughly) 10%.

- In Table 3, the flow-based formulation is the only approach that is able to solve all instances within the given time limit. Although possessing the worst LP bound, this model mostly lead to very good (or even the best) runtimes which is mainly caused by the fact that it has the least complex general structure (especially in terms of the constraints). However, in some cases, also the subtour or the cutset approach show competitive results. Contrary to that, the advantages of Martin's formulation cannot be observed for this test set.

- In Table 4, none of the approaches is able to solve every instance anymore. For these more challenging instances, the cutset formulation is able to obtain the most optimal solutions among the four modelling frameworks. Moreover, especially for instances of type 4, it also shows the best computation times on average. Contrary to that, in all cases, instances of type 3 can still be tackled by the flow-based formulation in very good times.

As a general observation, the flow-based formulation is likely to compensate the bad LP bound for small and moderately-sized instances, whereas the cutset (and partly the subtour) approach become more and more important when the complexity of the instances increases.

Table 3: Average runtime of the flow, cutset, subtour, and Martin's formulation of the MCI problem applying instance reductions and generalized model reduction (average values of 10 instances of types 1 and 2)

				LP relaxation				Runtime (in seconds)			
$\|V\|$	Type	$\|\mathcal{C}\|$	$\|E^\star\|_\varnothing$	F	C	S	M	F	C	S	M
16	1	16	20.1	18.5	19.6	19.7	19.7	**0.4**	3.4	2.6	1.5
		32	25.2	23.4	24.4	24.6	24.6	**1.1**	3.7	3.7	3.0
		48	29.7	27.4	28.7	28.9	28.9	1.3	1.7	**1.2**	1.9
		64	32.5	30.0	31.3	31.6	31.6	**1.5**	2.2	2.3	3.6
		80	35.3	32.8	34.2	34.4	34.4	1.4	1.7	**1.3**	2.5
	2	16	23.4	22.3	23.0	23.2	23.2	**0.2**	0.4	0.4	0.6
		32	31.6	29.7	31.0	31.2	31.2	0.8	**0.7**	0.8	1.1
		48	37.1	34.6	36.2	36.4	36.4	0.9	0.7	**0.6**	1.0
		64	42.2	39.5	41.2	41.4	41.4	1.1	0.8	**0.6**	1.2
		80	45.5	42.8	44.3	44.5	44.5	1.6	0.9	**0.8**	1.4
18	1	18	22.8	21.1	22.1	22.3	22.3	**1.6**	16.7	19.4	7.2
		36	29.2	26.7	28.2	28.4	28.4	**2.1**	8.9	7.7	6.5
		54	32.8	30.0	31.5	31.9	31.9	**4.8**	7.2	9.2	10.4
		72	38.1	35.3	36.9	37.2	37.2	4.7	**4.4**	4.9	8.3
		90	40.4	37.5	39.2	39.5	39.5	3.8	3.6	**3.5**	5.3
	2	18	27.5	25.3	26.9	27.1	27.1	**0.5**	0.8	0.7	0.9
		36	36.7	34.5	35.9	36.2	36.2	1.1	1.1	**1.0**	1.5
		54	43.0	40.1	41.9	42.2	42.2	1.5	1.5	**1.4**	2.4
		72	48.3	44.8	46.9	47.2	47.2	**2.0**	**2.0**	2.1	3.2
		90	52.6	48.9	51.1	51.3	51.3	2.3	1.5	**1.3**	2.5
20	1	20	25.3	23.4	24.6	24.8	24.8	**1.8**	43.2	34.3	7.7
		40	33.0	30.3	31.9	32.1	32.1	**13.4**	82.1	97.1	172.9
		60	37.7	34.4	36.1	36.4	36.4	**9.2**	29.9	36.2	35.0
		80	42.7	39.0	41.1	41.5	41.5	**8.6**	21.7	25.1	35.7
		100	46.6	41.9	44.3	44.7	44.7	**15.9**	20.6	23.8	41.9
	2	20	31.5	29.6	30.9	31.1	31.1	**0.6**	1.7	1.6	1.8
		40	41.2	38.2	40.0	40.2	40.2	**1.7**	2.8	2.6	3.5
		60	49.1	45.5	47.6	47.9	47.9	11.4	**6.5**	20.3	13.8
		80	56.0	51.5	54.2	54.5	54.5	4.4	**3.7**	4.6	7.5
		100	60.9	56.2	58.8	59.2	59.2	6.9	**4.2**	4.7	7.6
22	1	22	28.7	26.2	27.8	28.2	28.2	**2.8**	104.5	68.3 [9]	24.0
		44	37.8	34.1	36.3	36.7	36.7	**42.6**	111.3	379.1	193.8
		66	44.5	40.6	43.2	43.6	43.6	**5.8**	27.2	26.0	28.5
		88	47.6	42.8	45.2	45.7	45.7	**182.8**	88.5 [9]	180.5 [9]	187.5 [9]
		110	52.6	47.6	50.5	51.1	51.1	**46.6**	61.6	107.6	180.9
	2	22	35.1	32.7	34.3	34.6	34.6	**1.4**	4.1	5.3	3.5
		44	46.8	42.6	45.0	45.5	45.5	**6.7**	14.0	17.2	16.5
		66	56.2	51.7	54.4	54.8	54.8	24.9	19.9	**17.4**	119.0
		88	61.8	56.4	59.6	60.2	60.2	20.2	**9.8**	22.3	31.0
		110	68.6	63.2	66.4	66.8	66.8	295.0	**70.8**	256.1	97.1

Table 4: Average runtime of the flow, cutset, subtour, and Martin's formulation of the MCI problem applying instance reductions and generalized model reduction (average values of 10 instances of types 3 and 4)

				LP relaxation				Runtime (in seconds)			
$\|V\|$	Type	$\|\mathcal{C}\|$	$\|E^\star\|_\varnothing$	F	C	S	M	F	C	S	M
16	3	16	19.2	18.1	18.8	18.9	18.9	**0.6**	4.7	6.8	3.0
		32	24.1	22.1	23.1	23.3	23.3	**1.8**	3.5	5.8	5.1
		48	26.4	23.9	24.9	25.1	25.1	**3.5**	6.8	14.5	14.3
		64	29.6	27.0	28.1	28.2	28.2	**4.0**	6.4	15.6	28.4
		80	33.2	30.3	31.5	31.7	31.7	**5.0**	**5.0**	10.8	24.6
	4	16	24.0	22.7	23.5	23.6	23.6	**0.3**	0.5	0.7	0.9
		32	30.0	27.9	29.2	29.3	29.3	**1.1**	1.2	1.5	2.4
		48	34.9	32.3	33.5	33.6	33.6	3.3	**2.1**	3.5	6.0
		64	38.1	34.8	36.0	36.1	36.1	4.9	**3.1**	7.2	13.4
		80	39.9	36.4	37.6	37.7	37.7	10.2	**4.8**	16.3	24.1
17	3	17	20.9	19.4	20.2	20.3	20.3	**1.4**	8.8	16.2	6.3
		34	25.5	23.1	24.2	24.4	24.4	**5.9**	14.6	41.1	34.1
		51	28.0	25.3	26.4	26.5	26.5	**9.8**	27.2	82.3	86.8
		68	30.8	27.7	28.8	29.0	29.0	**19.2**	70.7	206.5	199.8
		85	32.5	29.2	30.4	30.5	30.5	**59.6**	141.7	586.6	536.5 [9]
	4	17	25.9	24.0	25.2	25.4	25.4	**0.6**	0.9	1.2	1.6
		34	32.0	29.5	30.7	30.9	30.9	2.5	**1.8**	3.3	5.6
		51	36.7	33.7	34.7	34.9	34.9	6.7	**5.1**	20.4	25.9
		68	39.8	35.5	36.8	36.9	36.9	196.0	**39.8**	186.5	425.9
		85	42.6	37.6	38.8	39.0	39.0	613.3 [9]	**456.6**	1060.9	1761.0 [7]
18	3	18	21.9	20.3	21.1	21.3	21.3	**2.2**	21.3	435.2	15.9
		36	27.1	24.4	25.7	25.9	25.9	**6.5**	89.2	249.9	111.4
		54	30.8	27.7	29.0	29.2	29.2	**54.4**	154.0	213.9	217.7
		72	34.1	30.3	31.7	31.9	31.9	**123.5**	358.1	586.7 [9]	795.3 [8]
		90	36.3	32.6	33.9	34.1	34.1	**238.4**	157.8 [9]	763.1 [9]	754.0 [9]
	4	18	26.7	25.0	26.1	26.2	26.2	**0.7**	1.4	2.1	2.4
		36	34.9	32.1	33.3	33.5	33.5	**4.4**	4.5	7.5	14.7
		54	38.6	34.9	36.3	36.5	36.5	34.8	**19.9**	75.8	86.1
		72	42.2	37.6	38.9	39.1	39.1	676.2 [8]	**202.3** [9]	463.9 [8]	1481.6 [8]
		90	44.6	39.4	40.8	40.9	40.9	919.9 [6]	**651.7** [8]	1594.6 [5]	2334.9 [3]

So far, our instances possessed at least as many clusters as vertices (and in most cases, actually, even much more clusters than vertices). In order to also study the performance of the different modelling frameworks for another category of instances, we now focus on the testset *kind 2*, see Table 5. Compared to the previous numerical results, these instances (having more vertices than clusters) considerably differ in their computational behavior. More precisely, the most interesting observations are given by:

Table 5: Average runtime of the flow, cutset, subtour, and Martin's formulation of the MCI problem using only instance reduction or instance reductions followed by generalized model reduction (average values of 10 instances of *kind 2*)

						%age red IR		Runtime (in seconds)							
								IR				IgMR			
$\|V\|$	$\|\mathcal{C}\|$	p	k	$\|\mathcal{C}\|_\varnothing$	$\|E^\star\|_\varnothing$	clus	vert	F	C	S	M	F	C	S	M
20	10	2	3	9.3	24.8	11.5	37.5	3.5	0.5	**0.4**	1.1	6.8	0.5	**0.4**	1.2
			4	9.8	24.2	5.1	32.5	15.8	0.7	**0.6**	**0.6**	11.4	**0.7**	2.4	**0.7**
			5	9.9	24.4	0.0	25.0	14.2	**1.0**	2.1	2.1	16.2	**1.0**	3.7	1.7
			6	10.0	22.7	1.0	18.0	19.6	5.7	**4.0**	4.1	14.8	5.2	4.0	**3.6**
30	10	3	3	10.0	38.0	2.0	30.7	19.8	4.6	**1.3**	2.0	22.2	4.8	**1.2**	1.9
			4	9.9	36.5	0.0	26.0	22.9	4.3	**2.8**	7.3	16.5	3.7	**3.1**	7.4
			5	10.0	35.1	0.0	21.3	22.7	17.1	23.3	**7.7**	21.2	13.5	14.8	**5.7**
			6	10.0	34.2	0.0	18.3	23.1	31.0	46.3	**11.2**	25.5	30.8	43.5	**11.3**
40	10	4	3	10.0	49.1	0.0	33.5	15.2	5.6	4.9	**3.0**	20.7	5.6	4.9	**3.0**
			4	10.0	48.2	0.0	23.5	45.4	76.8	65.7	**9.5**	44.0	74.2	71.9	**10.1**
			5	10.0	47.2	0.0	19.2	61.7	390.0	320.6	**42.6**	77.4	383.8	257.0 [9]	**44.1**
			6	10.0	45.4	0.0	16.0	74.1 [9]	477.9 [8]	732.9 [9]	**57.6**	56.2 [9]	477.8 [8]	901.7	**60.5**
50	10	5	3	10.0	61.3	0.0	31.2	40.4	29.0	24.4	**6.7**	34.9	29.3	24.8	**7.0**
			4	10.0	58.9	0.0	24.6	64.2	196.5	207.0	**41.6**	74.3	188.5	214.0	**42.4**
			5	10.0	57.1	0.0	18.6	156.5	816.3 [5]	1581 [6]	**116.3**	206.6	824.3 [5]	1735 [7]	**133.0**
30	15	2	3	14.3	40.9	10.6	32.0	17.3	0.7	**0.5**	0.7	17.6	0.9	**0.6**	0.8
			4	14.7	44.1	1.4	18.0	15.0	2.5	**1.2**	2.0	11.1	2.6	**1.2**	2.0
			5	14.7	41.4	0.7	19.3	21.0	4.0	**2.3**	3.8	18.3	3.7	**2.5**	3.4
			6	15.0	41.8	0.7	14.7	25.1	**5.8**	6.4	**5.8**	27.0	6.1	6.6	**5.7**
45	15	3	3	15.0	61.2	2.7	35.8	21.4	4.5	**1.6**	3.7	23.5	4.9	**1.8**	3.8
			4	14.9	62.6	0.0	26.2	28.5	7.4	8.0	**5.1**	37.9	7.4	7.8	**5.3**
			5	15.0	60.0	0.7	23.6	34.3	66.5	62.8	**12.8**	36.8	67.8	67.0	**14.4**
			6	15.0	59.0	0.0	19.1	48.9	150.6	149.3	**29.0**	49.9	132.9	149.2	**29.0**
60	15	4	3	15.0	81.5	1.3	33.7	40.2	20.3	10.2	**5.7**	44.0	20.8	10.4	**6.0**
			4	15.0	81.5	0.0	24.3	69.9	82.8	38.4	**11.1**	75.7	84.2	39.9	**12.7**
			5	15.0	79.1	0.0	19.8	82.9	398.1	283.9	**46.6**	85.4	397.4	303.5	**53.6**

- Contrary to the instances of *kind 1*, here we notice a substantial reduction in terms of the number of vertices caused by instance reduction rules (from [11]), whereas there is almost no (or only a very small) effect on the number of clusters. Without going too much into the details, most of these instance reductions (especially those that aim at decreasing the number of the clusters) also require specific "interactions" between the clusters to become applicable. Since we are now dealing with a rather small number of clusters, these conditions are not satisfied anymore, in general.

- Having a look at the runtimes with and without generalized model reduction, we do not observe any substantial differences. This is a clear hint that, very likely, the considered test set is not tailored for the application of the reduction rules presented in this paper. Again, the main reason is given by the fact that the required subset

cluster relations are very unlikely as there is only a few number of clusters (that are generated based on a relatively large number of vertices).

- In these experiments, it turns out that Martin's formulation is the only one that is able to solve all instances. Moreover, it mostly shows the best computation times. The flow-based formulation only fails to solve one single instance. However, it never proves to be the most efficient formulation, whereas the cutset and the subtour approach show competitive performances for at least some of the instances.

Note that we do not report about the intermediate step (i.e., the application of simple model reductions) in Table 5 since already the more general reduction concept did not lead to any significant benefits.

For many of the observations made so far, the argumentations were based on the conjectured effects of the (generalized) model reduction principles. More precisely, we mostly applied justifications dealing with the (expected) interaction of the clusters to explain the different numerical behavior of the various formulations. In order to examine the benefits or potentials of the proposed reduction methods in more details, we are now also reporting about the percentaged savings in terms of variables and constraints for any of the considered formulations. To this end, in Table 6, we selected a representative set of instances from *kind 1* and *kind 2* and stated the absolute number of constraints (columns 3-6) and variables (columns 7-10) after having applied the instance reductions from [11]. In the following columns, this value is considered as a level of 100% and forms a basis to compute the percentaged savings caused by simple model reduction (columns 11-18) and generalized model reduction (columns 19-26), respectively.

We would like to particularly point out the following results:

- Indeed, the flow-based approach offers the most reasonable trade-off between the numbers of variables and constraints. Contrary to that, the exponential-sized models (C and S) exhibit an "exploding" number of constraints, while Martin's formulation always possesses the most variables. These numerical data can be seen as clear indicators for the observation that the flow-based model is competitive in the majority of the cases, although it actually shows the worst results in terms of the LP bounds.

- For instances of *kind 1*, we can almost always notice a significant reduction in terms of constraints and variables (at least for those formulations having more than only the x-variables). These savings are particularly high for types 1-3, where rather large clusters can appear together with relatively small clusters, so that the required subset relations are easier to obtain, in general. As regards type 4, we know that the cluster sizes are in the smallest range, so that here we normally have the weakest reduction effects. However, also for these challenging instances, a level of up to (roughly) 25% of reduction is achievable.

- For instances of *kind 2*, our model reduction principles are only beneficial if the numbers of vertices and clusters are not too far away from each other. However, also in these cases, the effects are not comparably high as for instances of *kind 1*. Moreover, for the larger instances, we clearly see that no reduction at all is obtained. This relies on the fact that we have exponentially many possibilities to build a cluster, but only a very few of them are actually collected to define an instance.

Table 6: Average percentage of reduction, in terms of constraints and variables, caused by simple or generalized model reduction (applied after performing instance reductions)

		IR (#)								%age red IsMR								%age red IgMR							
		Cons				Var				Cons				Var				Cons				Var			
$\|V\|$	$\|\mathcal{C}\|$	F	C	S	M	F	C	S	M	F	C	S	M	F	C	S	M	F	C	S	M	F	C	S	M
kind 1																									
Type 1																									
15	15	536.4	12037.7	24391.2	4971.4	878.2	105.0	537.0	7864.2	35	84	84	52	36	0	32	55	43	86	86	59	44	0	39	62
	75	1448.9	11647.9	24040.2	10962.7	2059.2	105.0	1222.4	16360.4	44	88	87	64	50	0	46	68	63	94	94	80	69	0	63	84
Type 2																									
15	15	216.8	334.5	737.7	1019.1	342.0	100.8	248.5	1380.7	11	20	20	17	15	0	15	19	12	21	21	18	16	0	16	20
	75	922.2	1363.7	2995.0	4174.2	1104.9	105.0	722.7	5273.7	33	58	59	51	40	0	35	57	48	72	73	68	56	0	49	73
Type 3																									
15	15	533.2	12208.0	24739.4	5064.6	874.7	94.5	528.5	8030.9	29	70	70	42	30	0	26	44	33	71	71	46	33	0	30	48
	75	1768.2	13382.4	27697.1	13492.9	2521.3	105.0	1482.0	20098.8	40	81	80	57	46	0	42	61	48	83	82	63	53	0	49	67
Type 4																									
15	15	278.9	460.0	1018.7	1393.5	425.9	100.9	297.2	1876.4	2	3	3	3	5	0	5	4	2	3	3	3	5	0	5	4
	75	1359.5	2212.8	4902.1	6744.0	1684.0	105.0	1059.9	8670.9	16	26	26	23	19	0	17	25	17	27	27	24	20	0	18	26
kind 2, $k = 5$																									
20	10	194.6	552.8	1181.9	1097.8	319.8	84.5	224.5	1561.9	10	19	18	15	9	0	7	16	10	19	19	15	10	0	8	16
30	10	408.9	9209.7	18655.5	3583.1	793.0	209.7	537.1	5719.3	3	36	35	5	2	0	2	6	3	36	35	5	2	0	2	6
40	10	789.6	104832.0	210208.2	9451.0	1645.8	409.7	1081.6	15932.2	1	3	3	2	1	0	1	2	1	3	3	2	1	0	1	2
50	10	1123.4	750345.6	1501522.4	16414.0	2461.5	628.0	1610.3	28524.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	15	311.6	1181.7	2494.4	1869.2	555.9	169.8	397.5	2744.1	2	5	5	3	2	0	1	3	2	5	5	3	2	0	1	3
45	15	625.5	24247.3	48860.6	5667.3	1266.8	368.1	871.2	9157.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	15	1083.8	102824.0	206381.0	12481.8	2367.0	688.1	1604.0	21042.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8. Summary and conclusions

In this paper, we introduced several (M)ILP formulations of the MCI problem which differ in the constraints ensuring the existence of a spanning tree for each cluster. Moreover, two principles to reduce the size of the corresponding models are proposed. Besides the main theoretical statement concerning the strength of the LP relaxations, all presented new contributions are computationally tested and compared with each other and with those reductions known from the literature [10, 11]. The tests are performed by means of two kinds of randomly generated instances covering problems with (much) more clusters than vertices and the opposite situation.

In detail, we can state

- The proposed formulations perform (very) differently, but no clear trend can be observed. However, we underline the fact that any of the new (M)ILP models for the MCI problem proved to be efficient for specific subsets of the instances attempted. More precisely, the flow-based formulation (having a polynomial number of variables and constraints) seems to be the most general approach, as it leads to good (or even the best results) for a large number of testsets (especially for instances of *kind1*). In many cases, its favorably low complexity can compensate the fact that it possesses the worst LP bounds. On the other hand, if the number of vertices is not too large, then the subtour and cutset models (with an exponential number of restrictions) are competitive. The advantages of Martin's formulation (having a (higher) polynomial number of variables and constraints, but also a good LP bound) become visible for instances of *kind 2*, containing a large number of vertices (compared to the number of clusters) so that reductions usually fail to achieve significant benefits.

- The application of model reduction is beneficial, in general. In many cases, already the use of simple model reduction (requiring less efforts in comparison to the generalized model reduction) leads to remarkable savings. Especially for instances of *kind 1*, the combination of the instance reduction rules followed by the generalized model reduction yields the best results. However, even for those cases (see *kind 2*), where no model reduction could be observed, the additional computational efforts (to search for possible reductions) are almost neglectable so that we recommend to always apply the generalized model reduction before solving an instance.

Our future research in this respect will be directed to further decrease the numbers of variables and constraints and to strengthen the models by supplementary strong conditions. Moreover, the investigation of powerful heuristics with reasonably good (and proven) approximation guarantee is a crucial task to also tackle much larger instances in a sufficiently good manner. We further mention that the formulation-dependent behavior of the considered instances has to be investigated in more details (also from a theoretical point of view), e.g., to obtain more insights to the specific properties that led to the surprisingly good results of Martin's model for *kind2* instances.

Acknowledgments

This work is supported in parts by a scholarship of the Governmental Scholarship Programme Pakistan – DAAD/HEC Overseas and by the German Research Foundation (DFG) in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing (HAEC)".

References

[1] Agarwal, D., and Araujo, J.-C. S., Caillouet, C., Cazals, F., Coudert, D., Pérennes, S.: Connectivity inference in mass spectrometry based structure determination. Lecture Notes in Computer Science: European Symposium on Algorithms, 289–300 (2013)

[2] Agarwal, D., Caillouet, C., Coudert, D., Cazals, F.: Unveiling Contacts within Macromolecular Assemblies by Solving Minimum Weight Connectivity Inference (MWC) Problems. Molecular & Cellular Proteomics 14(8), 2274–2284 (2015)

[3] Angluin, D., Aspnes, J., Reyzin, L.: Inferring social networks from outbreaks. Lecture Notes in Computer Science: International Conference on Algorithmic Learning Theory, 104–118 (2010)

[4] Chen, C., Jacobsen, H.-A., Vitenberg, R.: Algorithms based on divide and conquer for topic-based publish/subscribe overlay design. IEEE/ACM Transactions on Networking 24(1), 422–436 (2016)

[5] Chen, J., Komusiewicz, C., Niedermeier, R., Sorge, M., Suchý, O., Weller, M.: Polynomial-time data reduction for the subset interconnection design problem. SIAM Journal on Discrete Mathematics 29(1), 1–25 (2015)

[6] Chockler, G., Melamed, R., Tock, Y, Vitenberg, R.: Constructing scalable overlays for pub-sub with many topics. Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing, 109–118 (2007)

[7] Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Graduate School of Industrial Administration, Report 388 (1976)

[8] Chwatal, A.M., Raidl, G.R.: Solving the minimum label spanning tree problem by mathematical programming techniques. Advances in Operations Research, Volume 2011, Article ID 143732 (2011)

[9] Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. 4OR: A Quarterly Journal of Operations Research 8(1), 1–48 (2010)

[10] Dar, M. A., Fischer, A., Martinovic, J., Scheithauer, G.: A Computational Study of Reduction Techniques for the Minimum Connectivity Inference Problem. Advances in Mathematical Methods and High Performance Computing, Springer Advances in Mechanics and Mathematics, Chapter 7, pp. 135–148 (2019)

[11] Dar, M. A., Fischer, A., Martinovic, J., Scheithauer, G.: An improved flow-based formulation and reduction principles for the minimum connectivity inference problem. Optimization 68(10), pp. 1963–1983 (2019)

[12] Du, D.-Z.: An optimization problem on graphs. Discrete Applied Mathematics 14(1), 101–104 (1986)

[13] Du, D.-Z., Miller, Z.: Matroids and subset interconnection design. SIAM Journal on Discrete Mathematics 1(4), 416–424 (1988)

[14] Du, D.-Z., Kelley, D.F.: On complexity of subset interconnection designs. Journal of Global Optimization 6(2), 193–205 (1995)

[15] Fan, H., Hundt, C., Wu, Y.-L., Ernst, J.: Algorithms and implementation for interconnection graph problem. Lecture Notes in Computer Science: Combinatorial Optimization and Applications, 201–210 (2008)

[16] Fan, N., Golari, M.: Integer programming formulations for minimum spanning forests and connected components in sparse graphs. Proceedings of the International Conference on Combinatorial Optimization and Applications, 613–622 (2014)

[17] Hosoda, J., Hromkovič, J., Izumi, T., Ono, H., Steinová, M., Wada, K.: On the approximability and hardness of minimum topic connected overlay and its special instances. Theoretical Computer Science 429, 144–154 (2012)

[18] Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society 7(1), 48–50 (1956)

[19] Magnanti, T.L., Wolsey, L.A.: Optimal trees. Handbooks in operations research and management science 7, 503–615 (1995)

[20] Martin, R.: Using separation algorithms to generate mixed integer model reformulations. Operations Research Letters 3, 119–128 (1991)

[21] Pop, P.C.: New models of the generalized minimum spanning tree problem. Journal of Mathematical Modelling and Algorithms 3(2), 153–166 (2004)

[22] Prisner, E.: Two algorithms for the subset interconnection design problem. Networks 22(4), 385–395 (1992)

[23] Supowit, K.J., Plaisted, D.A., Reingold, E.M.: Heuristics for weighted perfect matching. Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC '80), 398–419 (1980)

[24] Wang, G.-W., Zhang, C.-X., Zhuang, J.: Clustering with Prim's sequential representation of minimum spanning tree. Applied Mathematics and Computation 247, 521–534 (2014)

[25] Zhong, C., Miao, D., Fränti, P.: Minimum spanning tree based split-and-merge: A hierarchical clustering method. Information Sciences 181(16), 3397–3410 (2011)

[26] Zhong, C., Miao, D., Wang, R.: A graph-theoretical clustering method based on two rounds of minimum spanning trees. Pattern Recognition 43(3), 752–766 (2010)

Appendix A. Pseudo-code for the Instances obtained by managing the vertex frequency

Algorithm 1 Instance generation of type $n < m$

Input: n, p, and the upper bound k for the vertex frequency.
Output: $\mathcal{C}$

$V_i := \emptyset,\ \forall i \in I.$
$m = pn$
$l = k(p+1)$
while $\bigcup_{V_i \in \mathcal{C}} V_i \neq \{1, \ldots, m\}$ **do**
    $V_i := \emptyset,\ \forall i \in I.$
    $L := \{1, \ldots, n\}$
    for all $i \in \{1, \ldots, m\}$ **do**
        Choose a random integer q from $\{1, \ldots, k\}$
        Randomly pick $S \subseteq L$ such that $|S| = q$.
        for all $j \in S$ **do**
            Add vertex i to cluster V_j
            if $|V_j| = l$ **then**
                $L = L \setminus \{j\}$
            end if
        end for
    end for
    if $\exists V_i \in \mathcal{C}$ such that $|V_i| < 2$ **then**
        $\mathcal{C} = \mathcal{C} \setminus V_i$
    end if
end while