Technische Universität Dresden
Institute of Numerical Mathematics

An Improved Flow-based Formulation and Reduction Principles for the Minimum Connectivity Inference Problem

Muhammad Abid Dar
Andreas Fischer
John Martinovic
Guntram Scheithauer

Technical Report
MATH-NM-05-2017
October 2017

An Improved Flow-based Formulation and Reduction Principles for the Minimum Connectivity Inference Problem[1]

Muhammad Abid Dar
Andreas Fischer
John Martinovic
Guntram Scheithauer

Institute of Numerical Mathematics
Faculty of Mathematics
Technische Universität Dresden
01062 Dresden, Germany

Abstract. The Minimum Connectivity Inference (MCI) problem represents an $\mathcal{NP}$-hard generalisation of the well-known minimum spanning tree problem and has been studied in different fields of research independently. Let an undirected complete graph and finitely many subsets (clusters) of its vertex set be given. Then, the MCI problem is to find a minimal subset of edges so that every cluster is connected with respect to this minimal subset. Whereas, in general, existing approaches can only be applied to find approximate solutions or optimal edge sets of rather small instances, concepts to optimally cope with more meaningful problem sizes have not been proposed yet in literature. For this reason, we present a new mixed integer linear programming formulation for the MCI problem, and introduce new instance reduction methods that can be applied to downsize the complexity of a given instance prior to the optimisation. Based on theoretical and computational results both contributions are shown to be beneficial for solving larger instances.

Keywords. Minimum connectivity inference, reduction rules, mixed integer linear programming model, subset interconnection design

1. Introduction

The Minimum Connectivity Inference (MCI) problem is a discrete optimisation problem whose decision version is $\mathcal{NP}$-complete [1,2]. The MCI problem has been studied in several areas and under different names. Before providing details on such aspects and on our contributions, we start with the definition of the problem.

Let $G = (V, E)$ denote a simple, undirected, and complete graph with vertex set $V = \{1, \ldots, m\}$ and edge set $E = \{e = \{j, k\} \mid j \neq k \text{ and } j, k \in V\}$. Moreover, let

$$\mathcal{C} = \left\{ V_i \subseteq V \;\middle|\; \bigcup_{i \in I} V_i = V,\ 1 < |V_i| \text{ for all } i \in I \right\}$$

denote a collection of subsets of V called *clusters*, where $I \neq \emptyset$ is a finite index set. Then, for the *instance* $(V, \mathcal{C})$, the MCI problem is to find a subset E^* of E with

[1]Updated version from October 28, 2017

{Muhammad_Abid.Dar, Andreas.Fischer, John.Martinovic, Guntram.Scheithauer}@tu-dresden.de

minimal cardinality while the induced subgraphs $G^*[V_i]$ of $G^* = (V, E^*)$ are connected for all $i \in I$. Note that $G^*[V_i]$ is the graph with vertex set V_i that contains exactly those edges of E^* having both vertices in V_i. Any such edge set E^* is termed *optimal edge set* or *solution* of the MCI problem for the instance $(V, \mathcal{C})$. The cardinality of an optimal edge set is called *optimal value* of the instance. Moreover, an edge set $E' \subseteq E$ is said to be *feasible* for $(V, \mathcal{C})$, if every cluster V_i, $i \in I$, is connected with respect to E'. Clearly, since $\mathcal{C}$ is a set, any two clusters $V_{i_1}, V_{i_2} \in \mathcal{C}$ are different.

The above problem was considered for the design of vacuum systems in a Chinese journal [3] from 1976. This seems to be its first appearance, a reference to this paper can be found on Ding-Zhu Du's homepage [4] and in [5]. The name Subset Interconnection Design (SID) was used in [2] for the study of three related problems among which the MCI problem occurs. Later on, in [6], the problem was referred to as SID problem. It turns out that the Minimum Topic-Connected Overlay problem (see [1,7–9]) to establish scalable overlay networks is again nothing else than the MCI problem. Furthermore, the MCI problem was investigated for the design of reconfigurable interconnection networks and called Interconnection Graph Problem [10]. In the context of inference in underlying social networks, the MCI problem was dealt with as Network Inference problem [11]. Last but not least, we would like to mention the use of the MCI problem in structural biology to discover connections within macro-molecular assemblies in [12,13]. This is also the source of the name Minimum Connectivity Inference problem. From our point of view, among all the names, the latter shows best the aim of the problem under consideration. Sometimes, the MCI problem with nonuniform edge weights is also dealt with in the aforementioned references.

In addition to the study of applications, several interesting complexity results related to the MCI problem can be found in most of the papers cited above as well as in [2,14]. Moreover, polynomial heuristic algorithms were developed that, possibly under further conditions, yield some feasible (but generally not optimal) edge set whose cardinality can be bounded by a certain multiple of the cardinality of an optimal edge set, see [1,2,6].

Recently, a mixed integer linear programming (MILP) formulation of the MCI problem was suggested in [12]. This MILP formulation allows an exact solution of the problem up to a certain instance size (depending on the available computing power).

In this work, we propose an improved MILP formulation. On the one hand, this indeed enables us to solve larger instances of the MCI problem exactly. On the other hand, recall that the continuous relaxation of a MILP model yields a linear program (LP) whose optimal value provides a lower bound for the optimal value of the original MILP model. The continuous relaxation of the improved MILP formulation leads to a sharper (in our computations to a remarkably sharper) lower bound for the optimal value of the MCI problem. This is even useful for problems that do not allow an exact solution in reasonable time. To be specific we note that the quality of a feasible edge set can be evaluated by simply comparing its cardinality with the lower bound obtained from a continuous relaxation of the MCI problem. In this way, it may be possible to stop (heuristic) algorithms if a certain quality of the feasible edge set is reached. Evidently, the quality of the lower bound influences the significance of this approach.

To further enhance the benefits just shown we also propose some new instance reduction rules. Such rules may enable to reduce the size of an instance, i.e., the number $|V|$ of vertices, the number $|\mathcal{C}|$ of clusters, or both. For existing rules the reader is referred to [8], where also an incorrectness in rules from [9,10] is revealed. Last but not least, we compare the practical behaviour of the improved MILP formulation with

that of the original one in respect to several criteria. To this end we also propose a scheme to generate random instances of the MCI problem. Results on the influence of the new reduction rules on the practical behaviour are shown as well.

The paper is organised as follows. In Section 2, we review the MILP formulation from [12] and discuss our improved formulation. It is shown that the optimal value of the continuous relaxation of the improved MILP formulation is at least as good as the optimal value of the relaxation of the MILP model from [12]. Section 3 is devoted to the development of three new instance reduction rules. We prove the equivalence of the original and the reduced instance for these reduction rules. Subsection 3.3 shows how the reduction rules are implemented. In Section 4, we discuss several computational aspects of the improved MILP formulation (mostly compared to the existing MILP formulation) and of the new reduction rules by means of several types of randomly generated instances of the MCI problem. In particular, this concerns the improvement of the optimal value of the continuous relaxation of the MILP formulations and computing times.

For later use, we introduce some further notions. To this end, let $G' = (V', E')$ be a graph with $V' \subseteq V$ and $E' \subseteq E$. Then, a *path* in G' is the sequence of edges of E' connecting a sequence of pairwise distinct vertices of V'. To denote a path between v_{i_1} and v_{i_l}, simply the corresponding sequence of vertices is used, i.e., $(v_{i_1}, v_{i_2}, \ldots, v_{i_l})$ with pairwise distinct vertices $v_{i_t} \in V'$ $(t = 1, \ldots, l)$. The graph G' is said to be *connected* if there is a path between every pair of (distinct) vertices in V'. G' is called *connected component* of G if G' is a maximal connected subgraph of G. Any graph (V', T) with $T \subseteq E'$ is called *spanning tree* of G' if (V', T) is connected and if T is of minimal cardinality. Let us finally mention that we write $M \subset N$ to underline that M is a proper subset of N. Otherwise, $M \subseteq N$ is used.

2. Mixed integer linear programming formulations

The first MILP model of the MCI problem was given in [12] and called *flow-based formulation*. It is described in Subsection 2.1. Our improved formulation and its discussion follow in Subsection 2.2. The MILP formulations make use of some arc and edge sets. The *arc set* of the graph (V, E) is given by

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

Moreover, for a subset $U \subseteq V$ of vertices, let the corresponding sets of arcs and edges induced by U be defined as

$$A(U) = \{(j, k) \mid j, k \in U, j \neq k\} \quad \text{and} \quad E(U) = \{\{j, k\} \mid j, k \in U, j \neq k\}.$$

Finally, let $i \in I$ and $j \in V_i$ be arbitrarily given. Then, the sets

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

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

2.1. The original flow-based formulation

For an arbitrary instance $(V, \mathcal{C})$ of the MCI problem, the following MILP model was suggested in [12]. The idea behind this model is to use constraints which ensure that the vertices within each cluster are connected by a nonzero flow. To this end, nonnegative flow variables f_a^i were introduced with cluster index $i \in I$ and arc index $a \in A$. Moreover, within each cluster V_i, some vertex $r_i \in V_i$ is fixed which produces $|V_i| - 1$ units of flow, whereas each of the $|V_i| - 1$ vertices different from r_i consumes one unit of flow, see the equations in (2) below.

$$\min \quad \sum_{e \in E} x_e \tag{1}$$

$$\text{s.t.} \quad \sum_{a \in A_i^-(j)} f_a^i - \sum_{a \in A_i^+(j)} f_a^i = \begin{cases} |V_i| - 1, & \text{if } j = r_i, \\ -1, & \text{if } j \neq r_i, \end{cases} \qquad i \in I, j \in V_i, \tag{2}$$

$$f_a^i \leq |V_i| \cdot y_a, \qquad i \in I, a \in A, \tag{3}$$

$$y_{(j,k)} \leq x_e, \qquad e = \{j, k\} \in E, \tag{4}$$

$$y_{(k,j)} \leq x_e, \qquad e = \{j, k\} \in E, \tag{5}$$

$$y_a \in [0, 1], \qquad a \in A, \tag{6}$$

$$f_a^i \geq 0, \qquad i \in I, a \in A, \tag{7}$$

$$x_e \in \{0, 1\}, \qquad e \in E. \tag{8}$$

Due to some reasons related to consistency and for later use, the above MILP model has a slightly modified notation if compared to [12]. For discussion and analysis of the above and the following MILP model, let us consider any vector $x = (x_1, \ldots, x_{|E|})^\top \in \{0, 1\}^{|E|}$ and define the associated edge set

$$E(x) = \{e \in E \mid x_e = 1\}. \tag{9}$$

Conversely, any edge set $\hat{E} \subseteq E$ uniquely defines a vector $x \in \{0, 1\}^{|E|}$.

Proposition 2.1. *Let $(V, \mathcal{C})$ denote an instance of the MCI problem. If (f^*, x^*, y^*) is a solution of the MILP model (1) – (8), then $E(x^*)$ is an optimal edge set of the instance $(V, \mathcal{C})$. Conversely, if E^* is an optimal edge set, then there is a solution (f^*, x^*, y^*) of the MILP model with $E(x^*) = E^*$.*

Although there is no proof of this assertion in [12], we omit a proof here with reference to a related proof of Proposition 2.2 for our improved MILP model in Subsection 2.2.

In order to solve an instance $(V, \mathcal{C})$ based on the MILP model (1) – (8), our numerical experiments show that only very small instances can be solved, see Section 4 for details. Therefore, our aim in this paper is to significantly increase the size of instances which can be solved exactly. To achieve this we suggest in the following subsection how the above MILP model can be modified.

2.2. An improved flow-based formulation

We first provide the new MILP model for the MCI problem. Then, the correctness of the new model is shown. Thereafter, the differences to the original model (1) – (8) are explained and advantages of the new model are discussed. In particular, we prove that the continuous relaxation of the new model is at least as strong as (in fact, often even stronger than) that of the original model. Our new MILP formulation is given by

$$\min \quad \sum_{e\in E} x_e \tag{10}$$

$$\text{s.t.} \quad \sum_{e\in E(V_i)} x_e \geq |V_i| - 1, \qquad i \in I, \tag{11}$$

$$\sum_{a\in A_i^-(j)} f_a^i - \sum_{a\in A_i^+(j)} f_a^i = -1, \qquad j \in V_i \setminus \{r_i\},\ i \in I, \tag{12}$$

$$f_{(j,k)}^i + f_{(k,j)}^i \leq (|V_i| - 1)x_e, \qquad i \in I,\ e = \{j,k\} \in E(V_i), \tag{13}$$

$$f_a^i \geq 0, \qquad i \in I, a \in A(V_i), \tag{14}$$

$$x_e \in \{0,1\}, \qquad e \in E. \tag{15}$$

Remark 1. In general, the vector $f := (f_a^i)$ of the new model (10) – (15) contains much less variables than in the original formulation (1) – (8). For simplicity, we however make no distinction between the f-vectors of both models since, for theoretical purposes, any f-component not occurring in the new model can be added to this and just set to 0.

Proposition 2.2. *Let $(V,\mathcal{C})$ denote an instance of the MCI problem. If $(\widetilde{f},\widetilde{x})$ is feasible for the MILP model (10) – (15), then $\widetilde{E} = E(\widetilde{x})$ is a feasible edge set of the instance $(V,\mathcal{C})$ with the same objective value. Conversely, if $\widetilde{E}$ is a feasible edge set, then there is a feasible point $(\widetilde{f},\widetilde{x})$ of the MILP model with the objective value $|\widetilde{E}|$.*

Proof. In the first part of the proof we show that, for any feasible edge set $\widetilde{E} \subseteq E$ of the MCI instance $(V,\mathcal{C})$, there exists a feasible point $(\widetilde{f},\widetilde{x})$ of the MILP model (10) – (15). First of all, we define

$$\widetilde{x}_e = \begin{cases} 1, & e \in \widetilde{E}, \\ 0, & e \in E \setminus \widetilde{E}. \end{cases}$$

Thus, if $\widetilde{f}$ can be constructed so that $(\widetilde{f},\widetilde{x})$ is feasible for the MILP model (10) – (15), then the objective value of the MILP model coincides with that of the MCI instance.

Now, for every $i \in I$, the following steps i) – v) are performed to define the values $\widetilde{f}_a^i$ so that $(\widetilde{f},\widetilde{x})$ is feasible for the MILP model.

i) Let the edge set

$$\widetilde{T}_i = \widetilde{E} \cap E(V_i) = \{\{j,k\} \in \widetilde{E} \,|\, j,k \in V_i\}$$

be defined. Since $\widetilde{E}$ is a feasible edge set, $(V_i, E(V_i))$ is connected with respect to $\widetilde{T}_i$. Therefore, $(V_i, E(V_i))$ contains a spanning tree. In a slight abuse of notation, one of these spanning trees is denoted by its edge set $T_i \subseteq \widetilde{T}_i$. Note that,

thereby, conditions (11) are satisfied.

ii) To obtain a representation of T_i we choose an arbitrary root vertex $r_i \in V_i$ and assign a predecessor/ successor relation as follows. Let

$$S(r_i) = \{j \in V_i \,|\, \{r_i, j\} \in T_i\}$$

indicate the set of successors of r_i. We now start to define counters ν_j, $j \in V_i$, providing a 'distance' (number of edges) of $j \in V_i \setminus \{r_i\}$ to root node r_i within T_i. Naturally, we set $\nu_{r_i} := 0$. Moreover, let $p(j) := r_i$ indicate the predecessor of $j \in S(r_i)$ with respect to T_i, and let $\nu_j := 1$ be the corresponding distance. Furthermore, we define the set

$$\overline{T}_i = T_i \setminus \{\{r_i, j\} \,|\, j \in S(r_i)\}$$

of edges not yet considered. After this initialisation, the definition of the counters is continued in iii).

iii) While $\overline{T}_i \neq \emptyset$, choose a vertex $j \in V_i \setminus \{r_i\}$ which already got a predecessor, but not yet a set of successors, and define

$$\begin{aligned} S(j) &= \{k \in V_i \mid \{j,k\} \in \overline{T}_i\}, \\ \nu_k &= 1 + \nu_j, && k \in S(j), \\ p(k) &= j, && k \in S(j), \\ \overline{T}_i &= \overline{T}_i \setminus \{\{j,k\} \mid k \in S(j)\}. \end{aligned}$$

Repeat iii) until $\overline{T}_i = \emptyset$ is reached. Because of the properties of spanning trees, each vertex $j \in V_i \setminus \{r_i\}$ possesses a distance value ν_j greater than zero and less than $|V_i|$. Moreover, for at least one $j \in V_i$ we have $S(j) = \emptyset$.

iv) The definition of the values $\widetilde{f}^i_a$ is now done by the following procedure.
Repeat (a) – (d) until all ν-values are equal to zero.

(a) Choose any vertex $j \in V_i$ with maximal ν_j.

(b) If $S(j) = \emptyset$, then set $\widetilde{f}^i_{(j,k)} = 0$ for all $k \in V_i \setminus \{j\}$, and

$$\widetilde{f}^i_{(k,j)} = \begin{cases} 1, & k = p(j), \\ 0, & k \in V_i \setminus \{j, p(j)\}. \end{cases}$$

(c) If $S(j) \neq \emptyset$, then set $\widetilde{f}^i_{(j,k)} = 0$ for all $k \in V_i \setminus (\{j\} \cup S(j))$, and

$$\widetilde{f}^i_{(k,j)} = \begin{cases} 1 + \sum_{v \in S(j)} \widetilde{f}^i_{(j,v)}, & k = p(j), \\ 0, & k \in V_i \setminus \{j, p(j)\}. \end{cases}$$

(d) Set $\nu_j = 0$.

v) Finally, setting $\widetilde{f}^i_{(k,r_i)} = 0$ for $k \in V_i \setminus \{r_i\}$, all arcs got values $\widetilde{f}^i_a$ which satisfy conditions (12) – (14) by construction.

In the second part of the proof, let $(\widetilde{f}, \widetilde{x})$ be a feasible point of the MILP model (10) – (15). Clearly, according to (9), $\widetilde{x}$ induces the edge set $\widetilde{E} = E(\widetilde{x})$. It remains to show that the graph $(V_i, E(V_i))$ is connected for each cluster $i \in I$. Because of (11),

at least $|V_i| - 1$ edges belong to

$$\widetilde{E}_i = \{e \in E(V_i) \mid \widetilde{x}_e = 1\}.$$

We now show that $(V_i, \widetilde{E}_i)$ is a connected graph (and thus $(V_i, E(V_i))$ as well). To this end, let us assume the contrary. Then, there exists a connected component $S \subset V_i$ (in particular $S \neq V_i$) and $r_i \in S$ such that

$$\widetilde{f}^i_{(j,k)} = \widetilde{f}^i_{(k,j)} = 0, \qquad j \in S, k \in V_i \setminus S$$

holds. (Otherwise, the conditions in (13) would lead to $\widetilde{x}_e = 1$ for at least one edge $e = \{j, k\}$ between S and $V_i \setminus S$.) In particular, we have

$$\widetilde{f}^i_{(r_i,k)} = \widetilde{f}^i_{(k,r_i)} = 0, \qquad k \in V_i \setminus S.$$

Using this and (12), we obtain that

$$\begin{aligned}
1 - |S| &= \sum_{j \in S \setminus \{r_i\}} \left(\sum_{a \in A_i^-(j)} \widetilde{f}^i_a - \sum_{a \in A_i^+(j)} \widetilde{f}^i_a \right) \\
&= \sum_{j \in S \setminus \{r_i\}} \left(\sum_{k \in S \setminus \{j\}} \widetilde{f}^i_{(j,k)} - \sum_{k \in S \setminus \{j\}} \widetilde{f}^i_{(k,j)} \right) \\
&= \sum_{j \in S \setminus \{r_i\}} \left(\widetilde{f}^i_{(j,r_i)} - \widetilde{f}^i_{(r_i,j)} \right) \\
&= \sum_{j \in V_i \setminus \{r_i\}} \left(\widetilde{f}^i_{(j,r_i)} - \widetilde{f}^i_{(r_i,j)} \right),
\end{aligned}$$

holds. By means of (12), we also get

$$1 - |V_i| = \sum_{j \in V_i \setminus \{r_i\}} \left(\sum_{a \in A_i^-(j)} \widetilde{f}^i_a - \sum_{a \in A_i^+(j)} \widetilde{f}^i_a \right) = \sum_{j \in V_i \setminus \{r_i\}} \left(\widetilde{f}^i_{(j,r_i)} - \widetilde{f}^i_{(r_i,j)} \right).$$

Since $|S| < |V_i|$, this is a contradiction. Hence, $(V_i, \widetilde{E}_i)$ is connected for every $i \in I$. Consequently, $\widetilde{E} = \widetilde{E}(\widetilde{x})$ is a feasible edge set. □

Remark 2. The main differences of both MILP models are as follows.

a) In the new model, the y-variables are completely removed which saves $|A| = |V|(|V| - 1)$ continuous variables and $2|V|(|V| - 1)$ constraints since conditions (4) – (6) do not occur any longer. Moreover, instead of the conditions in (3), the new conditions in (13) come into play.

b) Those $|I|$ equations in (2) which belong to $j = r_i$ (for any fixed $i \in I$) are removed since they depend linearly on the equations for $j \in V_i \setminus \{r_i\}$. This can be seen by just summing up all the latter equations. Thus, in the new model, (12) replaces (2) from the original formulation.

c) The conditions in (11) do not appear in the original model. For an optimal edge set E^* of an instance $(V, \mathcal{C})$ we know that, for any $i \in I$, the graph $(V_i, E^* \cap E(V_i))$ contains a spanning tree. Each spanning tree in this graph has $|V_i| - 1$ edges, so that condition (11) is satisfied for any x^* associated to an optimal edge set of the instance.

It is noteworthy that already any $(\widetilde{f}, \widetilde{x})$ satisfying conditions (12) – (15) yields a feasible edge set $E(\widetilde{x})$ of the corresponding MCI instance. The next theorem shows that the optimal value of the linear relaxation of the new MILP model, even without the conditions in (11), is never smaller than the optimal value of the relaxed original MILP model. Later on, we provide an instance of the MCI problem for which the relaxation of the new MILP formulation without the conditions in (11) has a strictly larger optimal value than the relaxation of the original model does. Moreover, if we consider the new MILP model, i.e., with the conditions in (11), then the continuous relaxation might be further strengthened.

Theorem 2.3. *Let an instance $(V, \mathcal{C})$ of the MCI problem be given. Moreover, let $\tilde{z}$ denote the optimal value of the LP*

$$\min \quad \sum_{e \in E} x_e \qquad \text{s.t.} \quad (2) \text{ – } (7), \quad 0 \leq x_e \leq 1 \text{ for all } e \in E \tag{16}$$

and $\hat{z}$ the optimal value of the LP

$$\min \quad \sum_{e \in E} x_e \qquad \text{s.t.} \quad (12) \text{ – } (14), \quad 0 \leq x_e \leq 1 \text{ for all } e \in E. \tag{17}$$

Then, it holds that $\hat{z} \geq \tilde{z}$.

Proof. It suffices to show that the LP (16) has a feasible point $(\tilde{f}, \tilde{x}, \tilde{y})$ with objective value

$$\hat{z} = \sum_{e \in E} \tilde{x}_e. \tag{18}$$

Let $(\hat{f}, \hat{x})$ denote a solution of the LP (17) and define $(\tilde{f}, \tilde{x}, \tilde{y})$ by (keeping in mind Remark 1)

$$\tilde{f} = \hat{f}, \quad \tilde{x} = \hat{x}, \tag{19}$$

and

$$\tilde{y}_{(j,k)} = \hat{x}_{\{j,k\}}, \qquad \{j,k\} \in E. \tag{20}$$

Constraints (13) and (14) imply that

$$0 \leq \max\{\hat{f}^i_{(j,k)}, \hat{f}^i_{(k,j)}\} \leq (|V_i| - 1) \cdot \hat{x}_{\{j,k\}}, \qquad i \in I,\ \{j,k\} \in E(V_i). \tag{21}$$

Thus, $(\widetilde{f}, \widetilde{x}, \widetilde{y})$ satisfies the conditions in (3). Taking into account (19) and (12), it follows further that the conditions in (2) are satisfied for any $i \in I$ and $j \in V_i \setminus \{r_i\}$.

For $j = r_i$ the conditions in (2) hold due to part b) of Remark 2. Moreover, thanks to (19) – (20), $(\tilde{f}, \tilde{x}, \tilde{y})$ fulfils (4) – (7), whereas $\tilde{x}_e \in [0, 1]$ for $e \in E$ follows from (19) and (17). Finally, $\tilde{x} = \hat{x}$ ensures (18). This completes the proof. □

Example 2.4. Consider the MCI instance $(V, \mathcal{C})$ given by $V := \{1, 2, 3, 4\}$ and the clusters $\mathcal{C} := \{V_1, V_2, V_3, V_4\}$ with

$$V_1 := \{1, 2, 3\}, \quad V_2 := \{1, 2, 4\}, \quad V_3 := \{1, 3, 4\}, \quad \text{and} \quad V_4 := \{2, 3, 4\}.$$

Obviously, the complete graph (V, E) has $|E| = 6$ edges. Moreover, any optimal edge set E^* of this instance contains four edges. Table 1 provides the optimal values of the following three relaxations.

- LP (16) is the continuous relaxation of the original MILP model (1) – (8).
- LP (17) is the continuous relaxation of the new MILP model (10) – (15) without the conditions in (11).
- LP (17) with the additional constraints in (11) is the continuous relaxation of the new MILP model (10) – (15).

The optimal values of the relaxations may depend on the choice of the vertices r_i ($i \in I$). Therefore, Table 1 shows the lowest and the largest optimal value for each of the three relaxations. The results in Table 1 demonstrate that already the new MILP

Table 1. Comparison of the optimal values of different continuous relaxations for Example 2.4

Continuous relaxation	(16)	(17)	(17) including (11)
Optimal value depending on $r_1, \ldots, r_4$	$1.\bar{3}$ – 2.0	2.5 – 3.0	4.0 – 4.0

model without the conditions in (11) may provide a stronger relaxation compared with the relaxation of the original MILP model. Moreover, as Table 1 shows, the conditions in (11) lead to a further strengthening. In our exemplary instance, the optimal value of the continuous relaxation (17) is even equal to the optimal value $|E^*|$ of the MCI instance.

Finally, let us mention that some more flow variables are removed from the new MILP model. For every cluster V_i, the root node r_i only sends $|V_i| - 1$ units of flow, but it does not consume any unit of flow. Therefore, the flow variables $\{f_a^i \mid a \in A_i^+(r_i)\}$, which are attached to r_i, are superfluous and removed.

3. Instance reduction techniques

In this section, we first present two known instance reduction rules for the MCI problem (Subsection 3.1) and three new rules (Subsection 3.2). If applicable, these rules remove some of the vertices, some of the clusters, or both from an MCI instance and generate a *reduced* MCI instance. Hence, the application of such a rule results in a smaller, possibly easier to solve, MCI instance whose MILP model requires fewer constraints and variables than the original one.

An instance reduction rule possessing the property that any optimal edge set of the reduced instance can be converted into an optimal edge set of the original instance by simply adding a well-defined subset of the removed edges is called an *exact* reduction rule, otherwise a *heuristic* one.

In what follows, $\mathcal{H} = (V, \mathcal{C})$ denotes some instance of the MCI problem that is used as input for a reduction rule. For any vertex $u \in V$, let $\mathcal{C}(u)$ denote the set of all those clusters which contain vertex u, i.e., $\mathcal{C}(u) = \{V_i \mid u \in V_i, i \in I\}$. Furthermore, $\mathcal{H}_u$ or $\mathcal{H}_U$, respectively, denotes an instance of the MCI problem, where the vertex u or, respectively, all vertices in U are removed from V and from the clusters in $\mathcal{C}$.

Sometimes, the reduction rules presented in Subsection 3.1 lead to an instance $\tilde{\mathcal{H}} = (\tilde{V}, \tilde{\mathcal{C}})$, where $\tilde{\mathcal{C}}$ contains a cluster with less than two elements. Then, according to [12], such clusters are simply removed before any further reductions or exact solution techniques (for the reduced instance) are applied. Further note that some reduction rules may provide two or more identical clusters. Since the collection $\tilde{\mathcal{C}}$ of clusters is a set only one of those identical clusters remains.

3.1. Reduction rules from literature (Rules 1 and 2)

We next review two of the exact instance reduction rules proposed in [8], which shall later be included in our computational comparisons. Other rules analysed in [8] are not of interest here because they deal with quite special situations.

Rule 1 (Lemma 3.4 in [8])**.** *Assume that there are $q+1$ vertices $u, v_1, \dots v_q \in V$, with q being a positive integer, such that*

- *$\mathcal{C}(u) \subseteq \mathcal{C}(v_i)$ holds for each $i = 1, \dots, q$, and*
- *$|V_j| \leq q + 3$ is satisfied for all $V_j \in \mathcal{C}(u)$.*

Then, any solution E_u^* of $\mathcal{H}_u$ can be converted into a solution of $\mathcal{H}$ by adding a single edge $e = \{u, v\}$ to E_u^*, where v is an arbitrarily chosen vertex from the set $\{v_1, \dots v_q\}$.

By Rule 1, a single vertex can be removed from the original instance. A similar rule was proposed in [9,10,12]. However, as shown in [8] this rule is only a heuristic one. If applicable, the next rule allows to remove several vertices and at least one cluster.

Rule 2 (Lemma 5.8 in [8])**.** *Assume that there is a cluster $V_i \in \mathcal{C}$ with $V_i = \{u, u_1, \dots, u_l\}$ such that $\mathcal{C}(u_j) \subseteq \mathcal{C}(u)$ holds for each $j = 1, \dots, l$. Then, any solution $E^*_{\{u_1,\dots,u_l\}}$ of $\mathcal{H}_{\{u_1,\dots,u_l\}}$ can be converted into a solution of $\mathcal{H}$ by adding the edge set $\{\{u, u_j\} \mid 1 \leq j \leq l\}$ to $E^*_{\{u_1,\dots,u_l\}}$.*

3.2. New reduction rules (Rules 3 – 5)

In order to formulate further reduction rules we need to introduce some additional notation. For a given instance $(V, \mathcal{C})$ of the MCI problem, we define a graph $\mathcal{G} = (I, \mathcal{E})$ with vertex set I and edge set

$$\mathcal{E} = \{\{j, k\} \mid V_j \cap V_k \neq \emptyset, j, k \in I, j \neq k\},$$

that means, each cluster defines a vertex, and two clusters are adjacent if they possess a non-empty intersection. For any $J \subseteq I$, let

$$\mathcal{G}[J] = (J, \mathcal{E}(J)) \quad \text{with} \quad \mathcal{E}(J) = \{\{i, j\} \mid V_i \cap V_j \neq \emptyset, i, j \in J, i \neq j\}$$

denote the corresponding induced subgraph of $\mathcal{G}$. Moreover, we define the set

$$V(J) = \bigcup_{j \in J} V_j.$$

For a particular cluster $V_i \in \mathcal{C}$, let

$$J_i = \{j \in I \mid V_j \subset V_i, j \neq i\}$$

collect the clusters which are subsets of V_i. Finally, let γ_i denote the number of *connected components* of $\mathcal{G}[J_i]$.

Lemma 3.1. *Let $(V, \mathcal{C})$ be an instance of the MCI problem, and let $J \subseteq I$ be given such that $\mathcal{G}[J]$ is connected. Suppose that a set E' of edges forms a subgraph $G' = (V, E')$ of G such that $G'[V_i]$ is connected for all $i \in J$. Then, the induced graph $G'[V(J)]$ is also connected.*

Proof. To prove this lemma we take two arbitrary elements v_s and v_t of $V(J)$ and show that they are connected in $G' = G'[V(J)]$. Let $v_s \in V_l$ and $v_t \in V_p$ for some $l, p \in J$ be fixed. Since J is a connected component of $\mathcal{G}$ there is a path of edges (of $\mathcal{G}$) between l and p. Let this path be $\mathcal{P}(l, p) = (l, u_1, u_2, \ldots, u_r, p)$ with $u_i \in J$ for all $i \in \{1, 2, \ldots, r\}$. Hence, the following statements hold: $V_l \cap V_{u_1} \neq \emptyset, V_{u_1} \cap V_{u_2} \neq \emptyset, \ldots, V_{u_r} \cap V_p \neq \emptyset$. Without loss of generality, we assume that all of these intersections are singletons, i.e., we have $V_l \cap V_{u_1} = \{v_0\}$, $V_{u_1} \cap V_{u_2} = \{v_1\}$,$\ldots$,$V_{u_r} \cap V_p = \{v_r\}$ for appropriate vertices $v_0, v_1, \ldots v_r \in V(J) \subseteq V$.

Since $v_s, v_0 \in V_l$ for $l \in J$ there exists a path $P(v_s, v_0)$ between v_s and v_0 in G'. Similarly, as $v_0, v_1 \in V_{u_1}$, we have a path $P(v_0, v_1)$ between v_0 and v_1 in G'. After a finite number of such steps, we end up with a concatenated path $P(v_s, v_t)$ between v_s and v_t in G'. □

Rule 3. *If there exists a cluster $V_i \in \mathcal{C}$ with $V(J_i) = V_i$ and $\gamma_i = 1$, then any solution of $\mathcal{H}' = (V, \mathcal{C} \setminus \{V_i\})$ is also an optimal edge set for $\mathcal{H}$.*

Proof. Let $E^*_{\mathcal{H}'}$ be a solution of $\mathcal{H}'$. Then, by Lemma 3.1, it forms a subgraph $G' = (V, E^*_{\mathcal{H}'})$ such that $G'[V(J_i)] = G'[V_i]$ is connected and the proof is complete. □

If, for a certain cluster $V_i \in \mathcal{C}$, the number of connected components of $\mathcal{G}[J_i]$ is greater than one, i.e., if $\gamma_i > 1$ holds, then we represent these connected components by the index sets $J_{i,k} \subset J_i$ for $k = 1, \ldots, \gamma_i$. Hence, we have

$$J_i = \bigcup_{k=1}^{\gamma_i} J_{i,k}, \quad J_{i,k} \cap J_{i,l} = \emptyset, \qquad k, l \in \{1, \ldots, \gamma_i\} \text{ with } k \neq l.$$

Rule 4. *Assume that there is a cluster $V_i \in \mathcal{C}$ with $V(J_i) = V_i$ and $\gamma_i > 1$. If there is no cluster V_p with $p \in I \setminus \{i\}$, $V_p \cap V(J_{i,k}) \neq \emptyset$ and $V_p \cap V(J_{i,l}) \neq \emptyset$ for any pair (k, l) with $k, l \in \{1, \ldots, \gamma_i\}$ and $k \neq l$, then any solution $E^*_{\mathcal{H}'}$ of $\mathcal{H}' = (V, \mathcal{C}')$ with $\mathcal{C}' = \mathcal{C} \setminus \{V_i\}$ can be converted into a solution of $\mathcal{H}$ by adding $\gamma_i - 1$ edges $\{u, v_k\}$ to $E^*_{\mathcal{H}'}$, where $u \in V(J_{i,1})$ and $v_k \in V(J_{i,k})$ for $k = 2, \ldots, \gamma_i$ are arbitrarily chosen vertices.*

Proof. Let $E^*_{\mathcal{H}'}$ be a solution of $\mathcal{H}'$, and let $G' = (V, E^*_{\mathcal{H}'})$ be the corresponding induced subgraph. Let $u \in V(J_{i,1})$ and $v_k \in V(J_{i,k})$ for $k = 2, \ldots, \gamma_i$ arbitrarily chosen but fixed. Then, we define $E' = \{\{u, v_k\} \mid k = 2, \ldots, \gamma_i\}$ and $E'' := E^*_{\mathcal{H}'} \cup E'$. Because of $E' \subset E''$ and by definition, the induced graph $G'[V_j]$, and hence $G''[V_j]$ (as the induced graph with respect to $G'' = (V, E'')$), is connected for all $V_j \in \mathcal{C}'$. To prove that E'' is a solution of $\mathcal{H}$ we show that E'' is a feasible solution of $\mathcal{H}$ (first part) and that it is optimal as well (second part).

For the first part it is sufficient to prove that, for the graph $G''(V, E'')$, the induced graph $G''[V_i]$ is connected. To show that this sufficient condition is satisfied let us take two arbitrary vertices $\widetilde{v}_1, \widetilde{v}_2 \in V_i$ and consider the following two cases.

Case 1: Let $\widetilde{v}_1, \widetilde{v}_2 \in V(J_{i,k})$ be chosen from one and the same connected component $J_{i,k}$ of $\mathcal{G}[J_i]$. Then, by Lemma 3.1, $G'[V(J_{i,k})]$ is connected and there exists a path between $\widetilde{v}_1$ and $\widetilde{v}_2$ in G' and, therefore, in G''.

Case 2: Let $\widetilde{v}_1 \in V(J_{i,k})$ and $\widetilde{v}_2 \in V(J_{i,l})$ be chosen from different connected components $l \neq k, k, l \in \{1, \ldots \gamma_i\}$. We consider only the sub-case with $k \neq 1$, $l \neq 1$, $\widetilde{v}_1 \neq v_k$, and $\widetilde{v}_2 \neq v_l$. (The remaining sub-cases are very similar.) Then, by Lemma 3.1, there exist paths $P(\widetilde{v}_1, v_k)$ and $P(\widetilde{v}_2, v_l)$ between the pairs of vertices $\widetilde{v}_1, v_k$ and $\widetilde{v}_2, v_l$, respectively, in G', and hence in G''. By assumption, we also have $\{u, v_k\} \in E''$ and $\{u, v_l\} \in E''$. Altogether, we have found a path $P(\widetilde{v}_1, \widetilde{v}_2) = P(\widetilde{v}_1, v_k) \cup \{u, v_k\} \cup \{u, v_l\} \cup P(\widetilde{v}_2, v_l)$ between $\widetilde{v}_1$ and $\widetilde{v}_2$ in G''.

Hence, in any of the two cases, we obtain that $G''[V_i]$ is connected.

For the second part, we prove that

$$|E''| \leq |E^*| \tag{22}$$

holds for any fixed solution E^* of $\mathcal{H}$. Let $G^*(V, E^*)$ be the graph induced by E^*, and let us define the set of edges

$$E_\delta = \{\{v, w\} \in E \mid v \in V(J_{i,k}),\, w \in V(J_{i,l}),\, k \neq l\}.$$

By assumption, there is no cluster V_p, $p \in I \setminus \{i\}$, such that $V_p \cap V(J_{i,k}) \neq \emptyset$ and $V_p \cap V(J_{i,l}) \neq \emptyset$ (for $k \neq l$) are satisfied. This means that $E_\delta \cap E(V_j) = \emptyset$ holds for all $V_j \in \mathcal{C}'$. By definition, we further have $E' \subseteq E_\delta$ leading to $E^*_{\mathcal{H}'} \cap E' = \emptyset$. Hence, we obtain $|E''| = |E^*_{\mathcal{H}'} \cup E'| = |E^*_{\mathcal{H}'}| + |E'|$. Similarly, E^* can be divided into two disjoint edge sets $E^* \setminus E_\delta$ and $E^* \cap E_\delta$. Therefore, we have $|E^*| = |E^* \setminus E_\delta| + |E^* \cap E_\delta|$, and (22) can be formulated as

$$|E^*_{\mathcal{H}'}| + |E'| \leq |E^* \setminus E_\delta| + |E^* \cap E_\delta|. \tag{23}$$

To prove the latter, we claim that

$$|E^*_{\mathcal{H}'}| \leq |E^* \setminus E_\delta| \quad \text{and} \quad |E'| \leq |E^* \cap E_\delta| \tag{24}$$

are valid. To show that the second inequality holds true let us suppose the contrary, i.e., $|E^* \cap E_\delta| < |E'| = \gamma_i - 1$. Then, there is a connected component $J_{i,k}$ such that the set of edges

$$\{\{v, w\} \in E^* \cap E_\delta \mid v \in J_{i,k},\, w \in J_{i,l},\, l \neq k\}$$

is empty. But this implies that $G^*[V_i]$ is not connected which gives a contradiction. To see that the first inequality in (24) is valid we notice that E^* is a solution of $\mathcal{H}$ and, as argued earlier, $E_\delta \cap E(V_j) = \emptyset$ holds for all $V_j \in \mathcal{C}'$. Therefore, $E^* \setminus E_\delta$ represents a feasible solution of $\mathcal{H}'$, whereas $E^*_{\mathcal{H}'}$ is an optimal edge set of $\mathcal{H}'$. Consequently, we have $|E^*_{\mathcal{H}'}| \leq |E^* \setminus E_\delta|$. Thus, both inequalities in (24) are valid so that (23) holds and, hence (22) follows. □

Rule 5. *Assume that $V_i \in \mathcal{C}$ is a cluster satisfying $V(J_i) \neq V_i$ and $\gamma_i = 1$. If*

$$V_i \cap V_k = \{v\}$$

holds for all $V_k \in \mathcal{C}(v) \setminus \{V_i\}$ and all $v \in V_i \setminus V(J_i)$, then any solution $E^*_{\mathcal{H}'}$ of the instance $\mathcal{H}' = (V, \mathcal{C} \setminus \{V_i\})$ can be converted into a solution of the instance $\mathcal{H}$ by adding the edges in $T \cup \{\{u, v\}\}$ to $E^*_{\mathcal{H}'}$, where T is the edge set of an arbitrarily chosen spanning tree for the vertex set $V_i \setminus V(J_i)$, and $u \in V(J_i)$ and $v \in V_i \setminus V(J_i)$ are arbitrarily chosen vertices.

Proof. The proof of this theorem is similar to that of the previous one. Let $E^*_{\mathcal{H}'}$ be a solution of $\mathcal{H}'$, and let $G' = (V, E^*_{\mathcal{H}'})$ be the corresponding induced subgraph. Moreover, we consider the set $E' = T \cup \{\{u, v\}\}$ of edges and define $E'' := E^*_{\mathcal{H}'} \cup E'$, and let $G''(V, E'')$ be the related graph induced by E''. First we show that E'' is a feasible solution of $\mathcal{H}$. For this let $v_1, v_2 \in V_i$ denote arbitrary vertices and consider the following cases.

Case 1: Let $v_1, v_2 \in V_i \setminus V(J_i)$, then there exists a path $P(v_1, v_2)$ with edges from T between v_1 and v_2 in G''.

Case 2: Let $v_1, v_2 \in V(J_i)$ be given. Then, by Lemma 3.1, there exists a path between v_1 and v_2 in $G' \subseteq G''$.

Case 3: Without any loss of generality, we assume that $v_1 \in V_i \setminus V(J_i)$ and $v_2 \in V(J_i)$ are given. Then, there exists a path $P(v_1, v)$ with edges from T between v_1 and v in G''. Similarly, there exists a path $P(v_2, u) \subseteq E^*_{\mathcal{H}'}$ between v_2 and u in G''. By taking the union $P(v_1, v) \cup \{\{u, v\}\} \cup P(v_2, u)$, we get a path between v_1 and v_2 in G''.

Hence, in any of the three cases, we obtain that $G''[V_i]$ is connected.

Now, we prove that

$$|E''| \leq |E^*| \tag{25}$$

holds for any fixed solution E^* of $\mathcal{H}$. Let $G^* = (V, E^*)$ be the graph induced by E^*, and let us define the edge set

$$E_\alpha = E(V_i \setminus V(J_i)) \cup \{\{u, v\} \in E(V_i) \mid u \in V_i \setminus V(J_i),\, v \in V(J_i)\}.$$

By assumption, it is clear that $E_\alpha \cap E(V_j) = \emptyset$ holds for all $V_j \in \mathcal{C}'$. Since, by definition, we further have $E' \subseteq E_\alpha$, we obtain $E^*_{\mathcal{H}'} \cap E' = \emptyset$. Hence, the equation $|E''| = |E^*_{\mathcal{H}'} \cup E'| = |E^*_{\mathcal{H}'}| + |E'|$ is valid. Similarly, E^* can be divided into two disjoint edge sets $E^* \setminus E_\alpha$ and $E^* \cap E_\alpha$. Therefore, we have $|E^*| = |E^* \setminus E_\alpha| + |E^* \cap E_\alpha|$, and (25) can be written as

$$|E^*_{\mathcal{H}'}| + |E'| \leq |E^* \setminus E_\alpha| + |E^* \cap E_\alpha|. \tag{26}$$

As in the proof of Rule 4, we claim two inequalities: $|E^*_{\mathcal{H}'}| \leq |E^* \setminus E_\alpha|$ and $|E'| \leq$

$|E^* \cap E_\alpha|$. At first, note that $|E'| = |V_i \setminus V(J_i)|$ is valid. To prove the second inequality we suppose the contrary, i.e., $|E^* \cap E_\alpha| < |E'| = |V_i \setminus V(J_i)|$. Then, there exists a vertex $v \in V_i \setminus V(J_i)$ which is isolated in the induced graph $G^*[V_i]$ implying that $G^*[V_i]$ is not connected. Since this is a contradiction we have shown that the second inequality is satisfied. To prove the first inequality we notice that E^* is a solution of $\mathcal{H}$ and, as argued earlier in this proof, $E_\alpha \cap E(V_j) = \emptyset$ holds for all $V_j \in \mathcal{C}'$. Consequently, $E^* \setminus E_\alpha$ represents a feasible solution of $\mathcal{H}'$, whereas $E^*_{\mathcal{H}'}$ is a solution of $\mathcal{H}'$. Therefore, the first inequality $|E^*_{\mathcal{H}'}| \leq |E^* \setminus E_\alpha|$ holds as well. Thus, (26) is satisfied and implies (25). □

3.3. Implementation of the instance reduction rules

The pseudocode of the instance reduction rules explained in the previous subsections will be given in Appendix A. For this purpose, the subsequent definition of the matrices A and S as well as some observation on the meaning of their entries are helpful.

As before, let $\mathcal{H} = (V, \mathcal{C})$ denote an instance of the MCI problem. Then, $\mathcal{H}$ can be described by the cluster-vertex incidence matrix $A \in \mathbb{Z}^{|I| \times |V|}$ with

$$A_{i,j} = \begin{cases} 1, & \text{if cluster } V_i \text{ contains vertex } j, \\ 0, & \text{otherwise,} \end{cases} \qquad i \in I, j \in V.$$

To implement Rules 1 and 2 let the matrix

$$R = A^\top A \in \mathbb{Z}^{|V| \times |V|}$$

be defined. For two vertices $u, v \in V$, the entry $R_{u,v}$ of matrix R provides the number of clusters which contain both u and v. Therefore and since $R_{u,u} \geq R_{u,v}$, it follows that $\mathcal{C}(u) \subseteq \mathcal{C}(v)$ is satisfied if and only if $R_{u,v} = R_{u,u}$ holds.

To implement Rules 3 – 5 the matrix

$$S = AA^\top \in \mathbb{Z}^{|I| \times |I|}$$

is defined. Obviously, $S_{i,j} = |V_i \cap V_j|$ holds for all $i, j \in I$ and, as a special case, we have $S_{i,i} = |V_i|$ for all $i \in I$. Therefore, $V_j \subseteq V_i$ holds if and only if $S_{i,j} = S_{j,j}$ is satisfied, and we obtain

$$J_i = \{j \in I \mid V_j \subset V_i\} = \{j \in I \setminus \{i\} \mid S_{i,j} = S_{j,j}\}.$$

With reference to the pseudocode in Appendix A, we finally mention that the proposed reduction rules run in polynomial time depending on $|V|$ and $|I|$.

4. Computational experiments

This section contains a description of the computational environment and the procedure of generating test instances for the MCI problem. Moreover, we report on numerical results obtained by our simulations, together with a respective discussion and conclusions.

4.1. Instance generation

We consider four types of instances which differ in the (average) size of the clusters and in the range the size is chosen from. More precisely, four *instance types* are characterised by the minimum and maximum cardinality of all clusters $V_i \in \mathcal{C}$ as follows:

$$\begin{array}{lrcl} \text{Type 1:} & 2 & \leq |V_i| \leq & m, \\ \text{Type 2:} & 2 & \leq |V_i| \leq & \lceil m/2 \rceil, \\ \text{Type 3:} & \lceil m/4 \rceil & \leq |V_i| \leq & m, \\ \text{Type 4:} & \lceil m/4 \rceil & \leq |V_i| \leq & \lceil m/2 \rceil, \end{array} \tag{27}$$

where m defines the cardinality $|V|$ of the vertex set of an instance. Then, for given m, $|\mathcal{C}|$, and for a given instance type, the procedure in Appendix B generates 50 random instances. The cardinality of each of the clusters $V_1, \ldots, V_{|\mathcal{C}|}$ is drawn randomly from the set of feasible integers in (27) according to the uniform distribution. Thereafter, the vertex set of each of these clusters is drawn from $\{1, \ldots, |V|\}$ with uniform distribution. If, during the generation of a cluster, a vertex is drawn a second time, then it is not used. The same is done if identical clusters occur. For our computations, $|V|$ is chosen equal to or greater than 10, whereas $|\mathcal{C}| \in \{|V|, 3|V|, 5|V|\}$ is used, details are shown in the tables in the next subsection.

4.2. Computational results

The primary aim of our tests consists in the comparison of the two MILP models (1) – (8) and (10) – (15), i.e., the original and the improved one, with respect to the total computational time needed to solve an MCI instance. Secondly, the behaviour of the presented reduction rules shall be analysed. Finally, we investigate the performance of a technique for computing a feasible (usually not optimal) edge set by means of a heuristic proposed in [12].

To this end, the original MILP model (in Table 2) and the improved MILP model (in Tables 2 – 5) are applied to MCI instances either without or with the use of the presented reduction rules. For the latter case, we first sequentially apply the rules presented in Section 3 before solving the reduced instance by one of the MILP models.

Within Tables 2 – 5, we use the following notation.

- The column **Type** shows the instance type of a row, according to (27).
- In each row, the column $|\boldsymbol{E^*}|_\varnothing$ provides the averaged optimal values of the instances that were solved within the time limit of 900 seconds (out of 50 random instance).
- In each row, the column $\boldsymbol{\Delta}$ shows the value

$$100 \times \left(\frac{|E^{\text{heu}}|_\varnothing}{|E^*|_\varnothing} - 1 \right),$$

  where $|E^{\text{heu}}|_\varnothing$ denotes the averaged cardinalities of edge sets obtained by the heuristic technique from [12] applied to all instances solved within the time limit.
- The (multiple) columns **Clusters removed** and **Vertices removed** show percentages of how many clusters and vertices, respectively, could be removed by means of reduction rules.
- The columns **R1**–**R2**, **R1**–**R5**, and **R3** refer to results obtained by applying

Rules 1–2, Rules 1–5, and Rule 3, respectively, before solving the instance. The columns **+R3**, **+R4**, or **+R5** show the *additional percentage* of reduction obtained by applying Rule 3, Rule 4, or Rule 5, respectively, if Rules 1–2, Rules 1–3, or Rules 1-4, respectively, were already applied.

- The multiple column **Runtime** shows the averaged runtimes for solving the 50 instances depending on the reduction rules employed. The column **NoR** provides the averaged runtime if no reduction rule is used at all. In some of the entries of the column Runtime a **bracketed number [x]** can be found below the average runtime. This means that x<50 instances are solved within the time limit of 900 seconds and the average is taken only over these x instances.
- The two MILP formulations are indicated by **oMILP** (the original model (1) – (8) proposed in [12]) and by **iMILP** (the improved model (10) – (15)).

Note that all measurements (average percentages, average runtimes) given in the tables are rounded. Since the time needed by the heuristic to find the approximate solution is less than a second for all the tested MCI instances, we do not include the corresponding times in our tables.

In our last table, Table 6, we compare the relaxations of the original and the improved MILP model with respect to their optimal values and run times. More precisely,

- The columns $\mathbf{oLP}_{\varnothing}$ and $\mathbf{iLP}_{\varnothing}$ show the optimal values of the original and of the improved MILP model, averaged over 50 instances according to the number of clusters $|\mathcal{C}|$ and the instance type (27) as given in the corresponding row.
- The columns **min(iLP-oLP)** and **max(iLP-oLP)**, respectively, present the minimal and the maximal difference between the optimal values of the improved and the original MILP relaxation out of the 50 instances.
- Finally, the double column **Runtime** shows the averaged time needed for solving the relaxations of the original and of the improved MILP model (within the columns **oLP** and **oLP**).

The solution of any MCI instance is based on the solution of the MILP models in Section 2 by means of CPLEX® (Version 12.6.3) on a PC with Intel® Xeon® processor X5670 at 2.93 GHz using 96 GB of memory. The preprocessing including the application of reduction rules and the call of the CPLEX® routine `cplexmilp` is done in MATLAB® Release R2016a. After 900 seconds, the solution of any instance is stopped.

Furthermore, the relation between the two MILP models with respect to their computational behavior becomes obvious in Table 2. Due to the worse performance of the original MILP model we only apply the improved model for instances with larger vertex sets, and present results for those instances which mostly could be solved within the time limit of 900 seconds. Therefore, we give separate results for the cases $|\mathcal{C}| = m$, $|\mathcal{C}| = 3m$, and $|\mathcal{C}| = 5m$ in Tables 3 – 5.

4.3. Observations and conclusions

Based on the obtained computational results we now provide several observations and derive conclusions.

i) *Comparison of the MILP models.*
As Table 2 clearly shows, all the tested types of MCI instances could be solved much faster if the improved MILP model is used instead of the original model from [12]. This holds regardless of the application of reductions rules. Moreover,

Table 2. Computational Results obtained for the original MILP model (oMILP) and the improved model (iMILP) with and without instance reduction rules

					Clusters removed (%)				Vertices removed (%)	Runtime (sec) NoR		Runtime (sec) R1–R5	
m	Type	$\|\mathcal{C}\|$	$\|E^*\|_\varnothing$	Δ	R1–R2	+R3	+R4	+R5	R1–R2	oMILP	iMILP	oMILP	iMILP
10	1	10	10.8	4.8	13.8	11.2	0.0	0.0	15.6	1.08	0.16	0.71	0.14
		30	16.1	14.2	0.4	29.6	0.0	0.0	0.4	1.21	0.36	0.70	0.23
		50	19.4	13.6	0.0	42.1	0.0	0.0	0.0	1.74	0.65	0.56	0.25
	2	10	13.0	2.5	5.8	3.2	0.0	2.6	19.0	0.50	0.07	0.27	0.08
		30	20.4	7.3	0.0	10.6	0.0	0.2	0.0	0.36	0.14	0.30	0.16
		50	24.8	8.1	0.0	26.4	0.0	0.2	0.0	0.57	0.22	0.41	0.20
	3	10	10.7	6.0	9.8	13.6	0.0	0.2	12.0	1.31	0.15	0.99	0.12
		30	15.5	13.4	0.0	29.6	0.0	0.0	0.0	2.36	0.42	1.41	0.27
		50	18.3	14.2	0.0	41.2	0.0	0.0	0.0	3.92	0.69	1.53	0.32
	4	10	12.9	3.9	2.0	2.4	0.0	0.2	14.0	0.46	0.08	0.30	0.09
		30	19.6	8.8	0.0	6.5	0.0	0.0	0.0	1.03	0.21	0.95	0.24
		50	22.4	8.9	0.0	15.0	0.0	0.0	0.0	1.51	0.33	1.29	0.34
12	1	12	14.0	6.7	4.0	11.8	0.0	0.0	4.8	2.77	0.24	1.73	0.22
		36	20.4	13.2	0.0	27.3	0.0	0.0	0.0	3.73	0.75	1.91	0.44
		60	24.5	15.6	0.0	35.6	0.0	0.0	0.0	6.65	1.36	2.53	0.57
	2	12	16.2	3.7	3.2	0.8	0.0	0.0	9.0	0.70	0.10	0.60	0.11
		36	26.0	7.9	0.0	7.6	0.0	0.0	0.0	1.07	0.22	1.12	0.24
		60	31.2	9.3	0.0	18.2	0.0	0.0	0.0	2.12	0.36	1.44	0.36
	3	12	13.6	8.0	5.2	13.2	0.0	0.0	6.5	3.56	0.24	2.53	0.20
		36	19.8	16.6	0.0	27.1	0.0	0.0	0.0	11.25	0.95	4.20	0.52
		60	23.5	16.6	0.0	36.3	0.0	0.0	0.0	13.88	1.68	4.58	0.73
	4	12	16.4	5.0	0.2	1.0	0.0	0.2	5.7	1.01	0.13	0.96	0.15
		36	25.0	9.7	0.0	2.8	0.0	0.0	0.0	2.12	0.35	2.11	0.39
		60	29.4	10.1	0.0	9.1	0.0	0.0	0.0	4.01	0.61	3.41	0.62
14	1	14	16.6	10.8	2.0	12.0	0.0	0.0	2.4	30.8 [49]	0.49	21.9 [49]	0.36
		42	24.7	17.2	0.0	23.9	0.0	0.0	0.0	36.7	2.0	20.2	1.4
		70	30.1	16.9	0.0	30.8	0.0	0.0	0.0	36.4	3.53	11.8	1.81
	2	14	19.8	4.2	1.6	0.0	0.0	0.3	5.0	1.43	0.15	1.32	0.17
		42	31.3	10.4	0.0	3.9	0.0	0.0	0.0	3.11	0.48	2.92	0.46
		70	38.0	12.0	0.0	10.9	0.0	0.0	0.0	5.82	0.92	4.61	0.80
	3	14	16.4	9.8	0.6	13.1	0.0	0.0	1.0	76.9 [48]	0.59	50.6 [49]	0.39
		42	22.7	18.0	0.0	24.1	0.0	0.0	0.0	303 [44]	4.83	148	2.78
		70	26.5	18.8	0.0	31.6	0.0	0.0	0.0	493 [17]	11.3	269 [44]	4.86
	4	14	19.8	7.0	0.1	0.0	0.0	0.0	1.4	3.66	0.19	3.69	0.22
		42	29.1	11.1	0.0	0.7	0.0	0.0	0.0	52.9	1.90	53.0	1.90
		70	33.4	13.1	0.0	2.5	0.0	0.0	0.0	324 [47]	4.75	297 [46]	6.06

note that some of the instances with $m = 14$ vertices could not be solved within the given time limit if the original MILP model was used. Due to this, only the improved MILP model is used in the further tables.

ii) *Dependency on instance types.*
The computational behavior is quite different with respect to the four types of MCI instances. As it can be seen in Tables 2 and 3, Types 1 and 3 are more time consuming in comparison to Types 2 and 4, if the number of clusters is small. This could mainly be caused by the larger number of variables and constraints in the MILP model. However, as Tables 4 and 5 show, MCI instances of Type 4 become more difficult to solve with increasing number of clusters (and vertices).

Table 3. Computational results obtained for the improved MILP model with and without instance reduction rules for $|\mathcal{C}| = m$

				Clusters removed (%)		Vertices removed (%)	Runtime (sec)	
m	Type	$\|E^*\|_\varnothing$	Δ	R1–R2	+R3	R1–R2	NoR	R1–R5
16	1	19.9	10.4	1.3	10.9	2.5	1.06	0.85
	2	23.2	6.5	0.9	0.1	3.4	0.25	0.25
	3	19.2	13.2	2.0	11.9	2.0	2.03	1.54
	4	23.4	9.5	0.0	0.0	0.3	0.39	0.40
18	1	22.8	13.5	0.8	9.4	1.3	2.24	1.89
	2	27.5	6.7	0.6	0.0	1.6	0.34	0.37
	3	22.1	14.4	0.0	11.7	0.0	5.41	4.32
	4	27.0	10.6	0.0	0.0	0.2	0.92	0.91
20	1	25.9	15.2	0.4	8.9	0.3	8.69	9.46
	2	31.1	9.1	0.3	0.0	1.4	0.63	0.66
	3	25.2	16.4	0.0	9.7	0.0	8.68	6.33
	4	31.2	12.0	0.0	0.0	0.0	1.65	1.64
22	1	29.4	16.7	0.1	8.5	0.2	17.3	10.5
	2	35.4	10.5	0.1	0.0	0.9	1.40	1.47
	3	28.4	18.1	0.0	10.8	0.0	26.9	18.4
	4	35.1	11.9	0.0	0.0	0.2	3.93	3.96
24	1	32.4	18.6	0.1	7.8	0.1	62.4	61.3 [47]
	2	39.5	11.4	0.2	0.0	0.3	1.97	2.01
	3	31.4	18.1	0.0	9.5	0.0	99.6 [47]	64.5 [48]
	4	38.5	13.2	0.0	0.0	0.0	6.55	6.55
26	1	35.8	18.5	0.1	6.6	0.1	97.2 [49]	62.6 [48]
	2	44.2	11.9	0.1	0.0	0.2	3.20	3.25
	3	34.4	17.7	0.0	10.1	0.0	201 [31]	141 [37]
	4	42.4	13.6	0.0	0.0	0.0	41.8 [46]	41.8 [46]

iii) *Dependency on the number of clusters.*
It can be noticed that a larger number of clusters may lead to less successful reductions by Rules 1 and 2. This is mainly caused by the fact that the corresponding rules require special properties of the given clusters so that the sufficient conditions of these rules are harder to satisfy, in general. Hence, increasing the number of clusters (for constant number of vertices) does not only lead to a more difficult instance itself, but also to less beneficial reductions. Consequently, the hardness/complexity of an instance (if associated to the numbers of variables and constraints in the MILP) usually grows much faster than the number of clusters.

iv) *Behavior of the reduction rules.*
Concerning the effects of the reduction rules, we can state that Rule 3 is the most successful one. Note also that the percentages given in the tables for Rule 3 are reductions in addition to those caused by Rules 1 and 2. However, the obtained ratios of reduction also depend on the size and the type of instances. Rules 1 and 2 lead to a decrease of the number of clusters and vertices for small m in the case $|\mathcal{C}| = m$. Rules 3 – 5 are designed for the removal of clusters. Thus, for these rules, we do not provide computational results for the elimination of vertices. It can be seen in Table 2 that Rules 4 and 5 do not provide significant reductions for our test instances. Therefore, Rules 4 and 5 are not applied to

Table 4. Computational results obtained for the improved MILP model with and without instance reduction rules for $|\mathcal{C}| = 3m$

				Clusters removed (%)	Runtime (sec)	
m	Type	$\|E^*\|_\varnothing$	Δ	R3	NoR	R1–R5
16	1	29.4	19.8	20.2	6.17	3.20
	2	37.0	12.3	2.0	1.27	1.36
	3	27.1	19.9	22.8	13.6	6.75
	4	34.4	12.3	0.5	3.51	3.71
18	1	33.8	20.5	20.0	15.0	8.85
	2	42.8	13.3	0.9	2.92	3.11
	3	30.4	20.2	21.3	94.9	41.5
	4	39.1	13.2	0.1	70.8	68.0
					[49]	[49]
20	1	38.5	22.3	18.9	63.5	24.9
	2	49.4	14.1	0.6	6.33	6.44
	3	35.9	19.8	19.8	212	154
					[30]	[40]
	4	44.9	14.9	0.1	191	183
					[31]	[30]
22	1	44.3	22.7	16.7	150	80.2
					[44]	[49]
	2	55.6	16.8	0.4	24.9	25.0
					[49]	[49]
	3	39.4	20.3	16.8	594	425
					[1]	[7]
	4	—	—	—	—	—

Table 5. Computational results obtained for the improved MILP model with and without instance reduction rules for $|\mathcal{C}| = 5m$

				Clusters removed (%)	Runtime (sec)	
m	Type	$\|E^*\|_\varnothing$	Δ	R3	NoR	R1–R5
16	1	35.0	20.2	29.9	9.75	3.46
	2	45.5	13.6	6.4	2.26	2.24
	3	32.3	19.9	28.4	45.8	9.93
	4	40.6	15.1	1.3	19.6	19.8
18	1	41.0	19.8	26.4	40.1	13.3
	2	52.3	14.4	3.8	4.7	4.6
	3	35.9	21.5	25.3	285	135
					[28]	[41]
	4	44.6	13.5	0.1	286	354
					[4]	[5]
20	1	46.5	22.4	24.8	133	67.5
					[43]	
	2	60.1	16.2	2.5	16.9	18.0
	3	42.2	20.7	22.9	342	308
					[5]	[19]
	4	51.0	9.8	0.1	434	434
					[1]	[1]

larger instance sizes. Moreover, since the effects of Rules 1 and 2 go down with increasing number of vertices and clusters, we only use Rule 3 in Tables 4 and 5 to show the percentage of reductions. In average, Rule 3 is able to significantly improve the runtime for instances of Type 3 and 4.

v) *Performance of the Heuristic.*
The results provided by the heuristic cannot be considered as a good approximation of an optimal edge set. Hence, if an (almost) exact solution of MCI instances is desired, then due to this observation, the necessity of developing im-

Table 6. Computational results on optimal values and runtimes of the continuous relaxations of the original and the improved MILP formulation for instances with $m = 30$ vertices

		Optimal value				Runtime (sec)	
$\|\mathcal{C}\|$	Type	$oLP_\varnothing$	$iLP_\varnothing$	min(iLP-oLP)	max(iLP-oLP)	oLP	iLP
30	1	9.2	38.6	26.2	33.5	9.97	2.39
	2	14.0	48.2	30.0	37.6	6.27	0.62
	3	6.8	36.2	26.5	32.3	12.18	2.77
	4	10.2	45.6	32.7	38.4	8.66	1.28
60	1	17.6	57.0	35.2	46.1	113.10	6.17
	2	26.3	73.9	41.0	52.3	77.78	2.76
	3	10.8	48.2	34.6	41.0	157.28	9.55
	4	15.7	61.4	42.7	49.2	124.21	4.83
90	1	24.0	69.2	41.6	50.6	361.64	9.65
	2	35.2	90.6	51.6	61.5	309.36	4.59
	3	13.2	55.1	38.4	45.1	565.27	18.25
	4	18.7	68.5	47.5	51.9	524.30	10.34

proved exact solution approaches becomes obvious to enlarge the size of solvable MCI instances. Nevertheless, the heuristic might be helpful both for this purpose or for some applications.

vi) *Quality of LP relaxation.*
Table 6 shows clearly that the relaxation of the improved MILP model provides much better lower bounds for the optimal value of the MCI instances than the relaxation of the original MILP model does. This does not only hold in average but also for each of the 50 instances corresponding to a row in the table. In addition to this, solving the relaxation of the relaxation of the improved MILP model is much faster than for the original model.

5. Final remarks

In this paper, we have shown how the MILP model of the MCI problem from [12] can be improved remarkably. The advantages of the new model could be demonstrated by computational experiments. Moreover, some new instance reduction rules have been proposed for the MCI problem. Now, somewhat larger and more complex MCI instances can be solved to optimality in a modest amount of time. In addition, under some conditions, reduction methods can be helpful. Moreover, the computational tests showed that the (relative) gap between the objective function values obtained by a heuristic technique and the exact solution need not be small. Therefore, future work should aim to further improve corresponding MILP formulations, which enables to optimally solve even more complex and larger instances of the MCI problem, and to provide tighter lower bounds. As the different results obtained for the four types of MCI instances suggest, instance type specific formulations can be meaningful. The same could be true for the development of appropriate reduction methods.

Since the MCI problem is known to be $\mathcal{NP}$-hard, another possible area of future research is the investigation of sufficient conditions which can be used to prove that an approximate solution (obtained by a heuristic) is optimal. Moreover, the construction of heuristics having a proved performance behavior is of high interest.

Funding

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] Chockler G, Melamed R, Tock Y, et al. Constructing scalable overlays for pub-sub with many topics. In: Gupta I, editor. Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing; New York. ACM; 2007. p. 109–118.

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

[3] Du DZ, Chen YM. Placement of valves in vacuum systems. Communication on Electric Light Source Technology. 1976;4:22–28 (in Chinese).

[4] Du DZ. Curriculum Vitae of Ding-Zuh Du ; 2017. `http://www.utdallas.edu/~dxd056000/`, accessed July 25, 2017.

[5] Du DZ. An optimization problem on graphs. Discrete Applied Mathematics. 1986; 14(1):101–104.

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

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

[8] Chen J, Komusiewicz C, Niedermeier R, et al. Polynomial-time data reduction for the subset interconnection design problem. SIAM Journal on Discrete Mathematics. 2015; 29(1):1–25.

[9] Hosoda J, Hromkovič J, Izumi T, et al. On the approximability and hardness of minimum topic connected overlay and its special instances. Theoretical Computer Science. 2012; 429:144–154.

[10] Fan H, Hundt C, Wu YL, et al. Algorithms and implementation for interconnection graph problem. In: Yang B, Du DZ, Wang CA, editors. Combinatorial Optimization and Applications; (Lecture Notes in Computer Science; Vol. 5165). Springer; 2008. p. 201–210.

[11] Angluin D, Aspnes J, Reyzin L. Inferring social networks from outbreaks. In: International Conference on Algorithmic Learning Theory; (Lecture Notes in Computer Science; Vol. 6331); Berlin. Springer; 2010. p. 104–118.

[12] Agarwal D, Araujo JCS, Caillouet C, et al. Connectivity inference in mass spectrometry based structure determination. In: European Symposium on Algorithms; (Lecture Notes in Computer Science; Vol. 8125); Berlin. Springer; 2013. p. 289–300.

[13] Agarwal D, Caillouet C, Coudert D, et al. Unveiling contacts within macromolecular assemblies by solving minimum weight connectivity inference (MWC) problems. Molecular & Cellular Proteomics. 2015;14(8):2274–2284.

[14] Du DZ, Kelley DF. On complexity of subset interconnection designs. Journal of Global Optimization. 1995;6(2):193–205.

[15] Hopcroft J, Tarjan R. Algorithm 447: efficient algorithms for graph manipulation. Communications of the ACM. 1973;16(6):372–378.

Appendix A. Pseudocode for Rules 1 – 5

The notation within the algorithms below is taken from Sections 1 – 3. Furthermore, by $E^*_{R(k)}$, with $k \in \{1, 2, 4, 5\}$, the corresponding sets of edges obtained by applying Rule k are denoted. Since, for Rules 1 and 2, singleton clusters or identical clusters may be generated, lines 8–12 in the corresponding algorithms avoid this.

Algorithm 1 Rule 1

Input: The cluster-vertex incidence matrix A, the sets $V, \mathcal{C}$, and I of the MCI instance $(V, \mathcal{C})$.

Output: The reduced MCI instance $(V', \mathcal{C}')$ and the set of edges $E^*_{R(1)}$.

1: Set $V' = V, I' = I, V'_i = V_i \ \forall i \in I', \mathcal{C}' = \mathcal{C}$ and $E^*_{R(1)} = \emptyset$
2: Compute $R = A^\top A$
3: **for all** $u \in V$ **do**
4: Compute $C = \{v \in V' \mid R_{u,v} = R_{u,u}, v \neq u\}$
5: **if** $C \neq \emptyset$ and $|V_j| \leq |C| + 3, \forall j \in \mathcal{C}'$ **then**
6: Remove u from V' and from all $V'_i, i \in \mathcal{C}(u)$.
7: $E^*_{R(1)} = E^*_{R(1)} \cup \{u, v\}$ where $v \in C$ is an arbitrarily chosen vertex
8: **for all** $V'_i \in \mathcal{C}'$ **do**
9: **if** $|V'_i| \leq 1$ **or** $\exists\ V'_j \in \mathcal{C}'$, $i \neq j$ with $V'_i = V'_j$ **then**
10: Remove i from I'
11: **end if**
12: **end for**
13: Set $\mathcal{C}' = \{V'_i \mid i \in I'\}$
14: **end if**
15: **end for**

Algorithm 2 Rule 2

Input: The cluster-vertex incidence matrix A, the sets $V, \mathcal{C}$, and I of the MCI instance $(V, \mathcal{C})$.

Output: The reduced MCI instance $(V', \mathcal{C}')$ and the set of edges $E^*_{R(2)}$.

1: Set $V' = V, I' = I, V'_i = V_i \ \forall i \in I', \mathcal{C}' = \mathcal{C}$ and $E^*_{R(2)} = \emptyset$
2: Compute $R = A^\top A$
3: **for all** $u \in V$ **do**
4: Compute $C' = \{v \in V' \mid R_{v,u} = R_{v,v}\}$
5: **if** $|C'| > 1$ **and** $\exists\ i \in \mathcal{C}(u)$ such that $V_i \subseteq C'$ **then**
6: Compute $V' = V' \setminus V_i$ and $V'_j = V'_j \setminus V_i, j \in I'$
7: $E^*_{R(2)} = E^*_{R(2)} \cup \{\{u, v\} \mid v \in V', v \neq u\}$
8: **for all** $V'_i \in \mathcal{C}'$ **do**
9: **if** $|V'_i| \leq 1$ **or** $\exists\ V'_j \in \mathcal{C}'$, $i \neq j$ with $V'_i = V'_j$ **then**
10: Remove i from I'
11: **end if**
12: **end for**
13: Set $\mathcal{C}' = \{V'_i \mid i \in I'\}$
14: **end if**
15: **end for**

For the next algorithms, observe that the determination of connected components of a graph can be done in linear time (in terms of the number of vertices and edges of the graph), see for instance [15].

Algorithm 3 Rule 3

Input: The cluster-vertex incidence matrix A, the sets $V, \mathcal{C}$, and I of the MCI instance $(V, \mathcal{C})$.

Output: The reduced MCI instance $(V, \mathcal{C}')$.

Set $I' = I$
Compute $S = AA^\top$
for all $i \in I$ **do**
    Compute $J_i = \{j \in I \setminus \{i\} \mid S_{i,j} = S_{j,j}\}$
    if $V(J_i) = V_i$ and $\mathcal{G}(J_i)$ is connected **then**
        Remove i from I'
    end if
end for
Set $\mathcal{C}' = \{V_i \mid i \in I'\}$

Algorithm 4 Rule 4

Input: The cluster-vertex incidence matrix A, the sets $V, \mathcal{C}$, and I of the MCI instance $(V, \mathcal{C})$.

Output: The reduced MCI instance $(V, \mathcal{C}')$ and the set of edges $E^*_{R(4)}$.

Set $I' = I$ and $E^*_{R(4)} = \emptyset$
Compute $S = AA^\top$
for all $i \in I$ **do**
    Compute $J_i = \{j \in I \setminus \{i\} \mid S_{i,j} = S_{j,j}\}$
    if $J_i \neq \emptyset, \gamma_i > 1$ and $V(J_i) = V_i$ **then**
        if $\nexists\ V_p,\ p \in I \setminus \{i\}$ with $V_p \cap V(J_{i,k}) \neq \emptyset$ and $V_p \cap V(J_{i,l}) \neq \emptyset$ for any pair $k, l \in \{1, \ldots, \gamma_i\}, k \neq l$ **then**
            Remove i from I'
            Set $E = \{\{u, v_k\} \mid u \in V(J_{i,1}),\ v_k \in V(J_{i,k}),\ k \in \{2, \ldots, \gamma_i\}\}$, where $|E| = |\gamma_i - 1|$
            $E^*_{R(4)} = E^*_{R(4)} \cup E$
        end if
    end if
end for
Set $\mathcal{C}' = \{V_i \mid i \in I'\}$

Algorithm 5 Rule 5

Input: The cluster-vertex incidence matrix A, the sets $V, \mathcal{C}$, and I of the MCI instance $(V, \mathcal{C})$.

Output: The reduced MCI instance $(V', \mathcal{C}')$ and the set of edges $E^*_{R(5)}$.

Set $I' = I$, $V' = V$ and $E^*_{R(5)} = \emptyset$
Compute $S = AA^\top$
for all $i \in I$ **do**
    Compute $J_i = \{j \in I \setminus \{i\} \mid S_{i,j} = S_{j,j}\}$
    if $J_i \neq \emptyset, \gamma_i = 1$ and $V(J_i) \subset V_i$ **then**
        if $S_{i,k} = 1, \forall k \in \mathcal{C}(u) \setminus \{i\}, u \in V_i \setminus V(J_i)$ **then**
            Remove i from I'
            Set $\hat{E} = \{\{u, v\} \mid u \in V_i \setminus V(J_i)\}$ for some fixed but arbitrarily chosen $v \in V(J_i)$, where $|\hat{E}| = |V_i \setminus V(J_i)|$
            $E^*_{R(5)} = E^*_{R(5)} \cup \hat{E}$
        end if
    end if
end for
if $\exists v \in V'$ such that $\mathcal{C}(v) \cap I' = \emptyset$ **then**
    Remove v from V'
end if
Set $\mathcal{C}' = \{V_i \mid i \in I'\}$

Appendix B. Pseudocode for the random generation of clusters

Algorithm 6 For generating a set of clusters $\mathcal{C} = \{V_i \mid V_i \subseteq V, l \leq |V_i| \leq u\}$ by using uniform distributions

Input: m and n (desired number of vertices and clusters), l and u (lower and upper bound for cardinality of clusters according to the desired instance type in (27)).

Output: The set $\mathcal{C}$ of clusters.

$\mathcal{C} = \emptyset$
while $\bigcup_{V_i \in \mathcal{C}} V_i \neq V$ **do**
    for $i \in \{1, 2, \ldots, n\}$ **do**
        $V_i = \emptyset$
        while $|V_i| = \emptyset$ **do**
            Choose $p \in \{x \mid x \in \mathbb{N}, l \leq x \leq u\}$ randomly
            Choose $S \subseteq \{1, 2, \ldots, m\}$ randomly so that $|S| = p$
            Set $V_i = S$
            if $V_i \in \mathcal{C}$ **then**
                $V_i = \emptyset$
            end if
        end while
        Set $\mathcal{C} = \mathcal{C} \cup \{V_i\}$
    end for
end while