Correlation Clustering Problem under Mediation

Zacharie Alès, Céline Engelbeen, Rosa Figueiredo

Correlation Clustering Problem under Mediation

Zacharie Ales
UMA, CEDRIC, ENSTA Paris, Institut Polytechnique de Paris, France

Céline Engelbeen
Laboratoire Quaresmi, ICHEC, Brussels, Belgium

Rosa Figueiredo
Laboratoire Informatique d'Avignon, Avignon Université, France

Abstract

In the context of community detection, Correlation Clustering (CC) provides a measure of balance for social networks as well as a tool to explore their structures. However, CC does not encompass features such as the mediation between the clusters which could be all the more relevant with the recent rise of ideological polarization.

In this work, we study Correlation Clustering under mediation (CCM), a new variant of CC in which a set of mediators is determined. This new signed graph clustering problem is proved to be NP-hard and formulated as an integer programming formulation. An extensive investigation of the mediation set structure leads to the development of an efficient enumeration algorithm for CCM. Computational experiments are presented on two sets of instances: signed networks representing voting activity in the European Parliament and random signed graphs.

Keywords: Accessible system, Correlation clustering, Enumeration algorithm, Signed graph, Structural balance

1 Introduction

Community detection is largely applied to understanding the structure of social networks. In the presence of a network with antithetical relationships (like/dislike, for/against, similar/different...) community detection can be modeled as correlation clustering (CC), a signed graph clustering problem introduced by Bansal et al. (2004) for document classification.

In a signed graph, the edges are labeled as either positive (+) or negative (-). The CC problem consists in partitioning the vertices of such a graph while minimizing disagreements, i.e., the total number of positive edges between the clusters plus the total number of negative edges inside the clusters. A weighted version of the problem was lately defined in Demaine et al. (2006).

The CC problem is related to the concept of structural balance introduced in the field of social network analysis (Heider, 1946; Cartwright and Harary, 1956). According to structural balance theory, the equilibrium of a social system is associated with the propensity of individual elements to be organized in groups avoiding conflictual situations. This concept is perfectly described by graph theory (Davis, 1967). A signed graph is structurally balanced if it can be partitioned into clusters, such that all positive (resp. negative) edges are located inside (resp. in-between) these modules.

Applications of the CC problem overtakes the area of community detection problems and also arise in system biology (DasGupta et al., 2007), portfolio analysis for risk management (Figueiredo and Frota, 2014; Harary, 2002), voting behavior (Arinik et al., 2017; Kropivnik and Mrvar, 1996), document classification (Bansal et al., 2004), surface detection in 3D images (Kolluri et al., 2004), and in the detection of embedded matrix structures (Figueiredo et al., 2011). Variants of the CC problem have been proposed and discussed in the literature. Some of them motivated by a redefinition of the concept of structural balance (Doreian and Mrvar, 2009) or by applications to community detection in unsigned graphs.

The recent rise of ideological polarization makes it harder to reach agreements across partisan lines (Abramowitz and Saunders, 2008). Mediation could allow productive exchanges in polarized signed networks. In this context, we study a new variant of CC in which a set of key-players, called *mediators*, is additionally identified. We apply the concept of positive mediation as introduced by (Doreian and Mrvar, 2009): a set of mediators must have *good* relations among themselves and with other individuals in the network. A *good* relation is determined by two parameters, α and β, defining the minimal proportion of negative and positive relations allowed, respectively, inside and outside the mediation set.

The aim of the *correlation clustering problem under mediation (CCM)* is to obtain a partition which includes a cluster formed by *mediators* and which minimizes the imbalance (as defined in original CC) of the remaining clusters.

The contributions of this paper are fourfold.

1. We introduce the CCM problem, a new variant of CC in which the definition of a mediator set is parametrized by two parameters.

2. We prove that CCM is NP-hard and formulate this problem as an integer linear programming model.

3. We provide two explicit enumeration algorithms for CCM which take advantage of properties of mediator sets to break symmetry in the search tree.

4. We present extensive computational results to compare the performance of these algorithms with the resolution of our formulation by CPLEX.

The paper is organized as follows. The next section is dedicated to a review of the works related to the CCM problem. We give the notations and the formal definition of this problem in Section 3 and prove its NP-hardness. We introduce an ILP formulation of the problem in Section 4. Section 5 is devoted to the enumeration algorithms. Computational experiments are given in Section 6. We finally conclude the paper in Section 7.

2 Related works

The review of the literature is divided in three sections: exact optimization methods for CC (Section 2.1), variants of CC (Section 2.2) and group selection problems treated from a network optimization point of view (Section 2.3.)

2.1 Exact methods for CC

A combinatorial branch-and-bound was proposed by Brusco and Steinley (2009) to solve instances with up to 21 vertices. An Integer Linear Programming (ILP) formulation based on the vertex clustering formulation of (Mehrotra and Trick, 1998) was also considered (see for example (Demaine et al., 2006; Arinik et al., 2017, 2020)). It was used in a branch-and-cut framework on complete graphs with up to 50 vertices (Arinik et al., 2020) and on non-complete ones with up to 400 (Arinik et al., 2017). In (Figueiredo and Moura, 2013) the two approaches are compared. The authors showed that the ILP approach could handle larger graphs and required less time for most of the benchmark instances.

2.2 Variants of CC

Recent CC variants can be divided in two groups: redefinition of the objective function and redefinition of the clustering constraints.

CC seeks a partition which minimizes the total number of disagreements. Doreian and Mrvar (2009) observe that this definition does not encompass some important features. For example, vertices which agree with hostile subgroups increase the imbalance of the graph according to this definition. The authors considere that such vertices are potential mediators which should have a positive effect on the balance. Consequently, they propose a relaxed definition of the objective as the maximum disagreement inside and among all clusters in the partition. The Relaxed Correlation Clustering (Figueiredo and Moura, 2013; Levorato et al., 2017; Arinik et al., 2017) (RCC) consider this objective. Local disagreement functions have also been studied (Kalhan et al., 2019; Puleo and Milenkovic, 2018). These objectives minimize the number of disagreements at each vertex of the graph. For example, in (Puleo and Milenkovic, 2018) the total disagreement at the worst-off vertex in the partition is minimized.

Eventually, motivated by network analysis applications defined on unsigned graphs, Veldt et al. (2018) introduce the Lambda Correlation Clustering (LambdaCC), a weighted version of CC in which the weight of the edges is either $\lambda \in [0, 1]$ or $1 - \lambda$.

The first CC variants which redefines the constraints is Motif Correlation Clustering (MotifCC) (Li et al., 2017). Also motivated by network analysis applications, MotifCC associates the signs of the graph to motif patterns rather than edges. This variant generalizes CC to the hypergraph setting where the order of the graph is defined by the size of the motifs considered. In Fair Correlation Clustering (FairCC) the partition must satisfy fairness constraints. In (Ahmadian et al., 2020), each vertex of the graph has a color associated and the color in the partition must be distributed according to a given property (Chierichetti et al., 2017; Bera et al., 2019; Ahmadian et al., 2019). Figueiredo and Moura (2013) defined the first version of CC with mediation following the discussions in (Doreian and Mrvar, 2009). Their definition of a mediator set was very restrictive and we show that the problem defined in Section 2.3 generalises it.

Different approaches have been considered to solve these problems. ILP formulations were introduced in (Figueiredo and Moura, 2013) for RCC. Approximation algorithms were proposed for LambdaCC and MotifCC (Veldt et al., 2018; Li et al., 2017; Gleich et al., 2018) as well as for FairCC (Kalhan et al., 2019; Puleo and Milenkovic, 2018). A simulated annealing was considered for MotifCC in Li et al. (2017) while Iterated Local Search methods were proposed for RCC (Levorato et al., 2017).

2.3 Group selection in social networks

Several works have been dedicated to the identification of a set of individuals playing a specific role in a network. These individuals can be named key players (Borgatti, 2006; Ortiz-Arroyo, 2010), influential vertices (Li et al., 2011), or mediators (Figueiredo and Moura, 2013).

The set of vertices can be selected through a global network optimization criteria or by ranking network elements according to an individual measure (e.g., vertex centrality (Borgatti, 2003)). We focus on the first approach as the second can not generally guarantee the optimality of the solution found (see examples in (Ortiz-Arroyo, 2010)).

The key players problem as introduced by (Borgatti, 2003), consists in selecting k vertices in a network that maximizes or minimizes the disruption of the residual network obtained by removing them. Different measures and heuristic procedures have been proposed in the literature for this problem (Borgatti, 2006; Ortiz-Arroyo, 2010). (Li et al., 2011) studied the problem of finding the set of key players controlling the bottlenecks of influence propagation in a social network. They named it as the $k-$mediators problem and proposed a three-steps heuristic to solve it. We refer the reader to other references in (Li et al., 2011) on works selecting vertex sets playing an important role in influence maximization.

None of these works considered exact methods even when the size of the networks is small (see for example (Borgatti, 2006)). The CCM defined in this work is based on the mediation concept described by Doreian and Mrvar (2009). It has only been treated once in the literature (Figueiredo and Moura, 2013) and for a very particular case where both parameters defining the feasibility of the mediator sets are set to 0.

3 Notation and problem definition

Let $G = (V, E)$ be an *undirected graph*, where V and E are the sets of vertices and edges, respectively. Consider a function $s : E \rightarrow \{+, -\}$ that assigns a *sign* to each edge in E. An undirected graph G together with a function s is called a *signed graph*, denoted here by $G = (V, E, s)$. An edge $e \in E$ is called negative if $s(e) = -$ and positive if $s(e) = +$. We note E^- and E^+ the sets of negative and positive edges in a signed graph, respectively. Let $n = |V|$.

The CC problem (Bansal et al., 2004) aims to find a partition of the vertices which minimizes the number of disagreements, that is positive edges between two clusters or negative edges inside a cluster. The sum of all disagreements is called the *imbalance* of a partition. In the weighed version of the CC problem, an extra function $w : E \to \mathbb{R}^+$ is added. In order to define the imbalance in that weighted case, let us introduce some extra notations.

For two subsets $S_1, S_2 \subseteq V$ and a sign $\sigma \in \{+, -\}$ we define $E^\sigma[S_1, S_2] = \{(i, j) \in E^\sigma : i \in S_1, j \in S_2,\ i \neq j\}$, $w^\sigma(S_1, S_2) = \sum_{(i,j)\in E^\sigma[S_1,S_2]} w_{ij}$ and $w^\sigma(S_1) = w^\sigma(S_1, S_1)$.

A *partition* of V is a division of V into non-overlapping and non-empty subsets. The *imbalance* $I(P)$ of a partition $P = \{S_1, S_2, \ldots, S_{|P|}\}$ is the weight of the negative arcs in the subsets and of the positive arcs between the subsets, i.e.,

$$I(P) = \sum_{1\leq i\leq |P|} w^-(S_i) + \sum_{1\leq i<j\leq |P|} w^+(S_i, S_j). \tag{1}$$

As stated by Bansal et al. (2004), CC consists in finding a partition which imbalance (1) is minimal. Let us denote this minimal value by CC(G).

We introduce a new variant of CC in which a set of vertices called *mediators* is identified and which minimizes the imbalance of the remaining vertices. We now state two properties that the set of mediators must satisfy.

Definition 1. *Let α in $\mathbb{R}^+$. A subset $S \subseteq V$ is α-feasible if $\alpha w^+(S) \geq w^-(S)$.*

Definition 2. *Let β in $\mathbb{R}^+$. A subset $S \subseteq V$ is $\beta-$feasible if $\beta w^+(S, V\backslash S) \geq w^-(S, V\backslash S)$.*

These definitions provide upper bounds on the negative weights inside (Definition 1) and leaving (Definition 2) the vertex set S. Note that if α and β are both equal to 0, the vertex set S only contains non-negative weights and the weight of each arc leaving S is also non-negative. These two constraints together lead to the definition of the mediator set.

Definition 3. *A subset $S \subseteq V$ is a mediator set if S is $\alpha-$feasible and $\beta-$feasible.*

We can now formally define the Correlation Clustering problem under Mediation.

Correlation Clustering problem under Mediation	
Input:	A signed graph $G = (V, E, s)$, non-negative arc weights $w \in \mathbb{R}_+^{|E|}$ and two scalars $\alpha, \beta \in \mathbb{R}_+$.
Output:	A partition $P = \{S_M, S_2, ..., S_{|P|}\}$ which minimizes the imbalance $I(P\backslash S_M)$ and such that S_M is a mediator set.

The Correlation Clustering with Positive Mediation (CCPM) problem introduced in Doreian and Mrvar (2009) and formalized in Figueiredo and Moura (2013) is a specific case of CCM in which $\alpha = \beta = 0$.

We now prove that CCM is NP-hard.

Lemma 1. *The CCM problem is NP-hard.*

Proof. We prove this result with a reduction from CC. Consider an instance I_{CC} of CC defined over a signed graph $G = (V, E, s)$ with an edge weight vector $w \in \mathbb{R}_+^{|E|}$. Let $G' = (V', E', s')$ be a signed graph and let $w' \in \mathbb{R}_+^{|E'|}$ be an edge weight vector defined as follows (see Figure 1):

- $V' = V \cup \{n+1, n+2, n+3\}$
- $E' = E \cup E^1 \cup E^2 \cup E^3$ with:
  - $E^1 = \{(n+1, n+3), (n+2, n+3)\}$,
  - $E^2 = \{(n+1, n+2)\}$,
  - $E^3 = \{(n+2, i) : i \in V\} \cup \{(n+3, i) : i \in V\}$.
- $s'_e = \begin{cases} s_e, & e \in E, \\ +, & e \in E^1, \\ -, & e \in E^2 \cup E^3. \end{cases}$
- $w'_e = \begin{cases} w_e, & e \in E, \\ M, & e \in E^1 \cup E^2, \text{ with } M = 1 + \sum_{e \in E} w_e, \\ -3M, & e \in E^3. \end{cases}$

Consider an instance I_{CCM} of CCM defined over the signed graph G' with $\beta = 1$ and $\alpha \in [0, 1]$. Let $P_{CC} = \{S_1, \dots, S_{|P|}\}$ be an optimal solution of I_{CC} and consider partition $P_{CCM} = \{\{n+1\}, \{n+2, n+3\}, S_1, \dots, S_{|P|}\}$. We now prove that P_{CCM} is an optimal solution of I_{CCM}. The value of the CCM problem associated with P_{CCM} is $IP(P_{CC} \cup \{n+2, n+3\}) = IP(P_{CC}) < M$.

Vertices $n+1, n+2$ and $n+3$ define a non-balanced cycle in G' (i.e., a cycle with an odd number of negative edges) composed of edges of weight M. As a consequence at least one of them must be in the mediator set in an optimal solution (otherwise the imbalance would be greater than or equal to M).

If vertex $n+2$ or $n+3$ is in the mediator set, a vertex in V can not be in the mediator set – as it would be α-infeasible – and it can not either be outside of the mediator set, as it would be β-infeasible. As a consequence, vertex $n+1$ is necessarily in the mediator set of an optimal solution. Moreover, no vertex in V can be in the mediator set as it would be β-infeasible.

Consequently, the mediator set of I_{CCM} is necessarily $\{n+1\}$. Thus, P_{CCM} is an optimal solution of I_{CCM}. □

In the next section, we formulate the CCM Problem as an Integer Linear Programming (ILP) model.

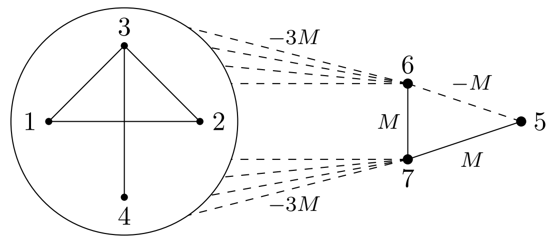


Figure 1: Example of the reduction from an instance of CC with 4 vertices to an instance of CCM with 7 vertices.

4 Mathematical formulation

ILP formulations have been successfully used in the literature for the resolution of clustering problems (Johnson et al., 1993; Mehrotra and Trick, 1996; Hansen and Jaumard, 1997; Agarwal and Kempe, 2008; Brusco and Steinley, 2009; Ales et al., 2016), including clustering problems defined on signed graphs (Figueiredo and Moura, 2013; Aref and Wilson, 2019). In this section, we introduce an ILP formulation for the CCM problem.

For each pair of distinct vertices i, j in V, we consider a binary variable x_{ij} equal to 1 if and only if i and j do not belong to the same cluster. Also, to each vertex $i \in V$ is associated a binary variable m_i equal to 1 if and only if i is a mediator. Note that in this formulation, each mediator vertex is represented as an isolated vertex. Finally, each pair of distinct vertices i, j is associated with two additional binary variables: t_{ij} equal to 1 if and only if both i and j are mediators; and z_{ij} equal to 1 if and only if at least i or j is a mediator.

$$
\begin{aligned}
\text{minimize} & \sum_{(i,j)\in E^-} w_{ij}(1-x_{ij}) + \sum_{(i,j)\in E^+} w_{ij}(x_{ij}-z_{ij}) && && (2)\\
\text{s.t. } & x_{jk} \leq x_{ij} + x_{ik}, && i\in V \;\; j,k\in V\backslash\{i\} \;\; j<k, && (3)\\
& m_i \leq x_{ij}, && i,j\in V \;\; i\neq j, && (4)\\
& m_i + m_j - 1 \leq t_{ij}, && i,j\in V \;\; i\neq j, && (5)\\
& t_{ij} \leq m_i, && i,j\in V \;\; i\neq j, && (6)\\
& m_i - m_j \leq z_{ij}, && i,j\in V \;\; i\neq j, && (7)\\
& z_{ij} \leq m_i + m_j, && i,j\in V \;\; i\neq j, && (8)\\
& \sum_{(i,j)\in E^-} w_{ij}t_{ij} \leq \alpha \sum_{(i,j)\in E^+} w_{ij}t_{ij}, && && (9)\\
& \sum_{(i,j)\in E^-} w_{ij}(z_{ij}-t_{ij}) \leq \beta \sum_{(i,j)\in E^+} w_{ij}(z_{ij}-t_{ij}), && && (10)\\
& x_{ij} = x_{ji} \in \{0,1\}, && i,j\in V \;\; i\neq j, && (11)\\
& z_{ij} = z_{ji} \in [0,1], && i,j\in V \;\; i\neq j, && (12)\\
& t_{ij} = t_{ji} \in [0,1], && i,j\in V \;\; i\neq j, && (13)\\
& m_i \in \{0,1\}, && i\in V. && (14)
\end{aligned}
$$

The triangle inequalities (3) ensure that if i is in the same cluster than j and k ($x_{ij} = x_{ik} = 0$), then vertices j and k are also in the same cluster ($x_{jk} = 0$). Constraints (4) establish that mediators are isolated. Constraints (5) and (6) ensure that $t_{ij} = m_i m_j$. Constraints (7) and (8) impose $z_{ij} = 1$ whenever $m_i + m_j \geq 1$. Note that, when $m_i + m_j = 2$, z_{ij} is set to 1 by the second term of the objective function. Constraints (9) and (10) ensure that the set of mediators is α and β-feasible, respectively. Remark that the expression $z_{ij} - t_{ij}$ is equal to 0 if and only if $m_i = m_j$. Consequently, for $\sigma \in \{-,+\}$, $\sum_{(i,j)\in E^\sigma} w_{ij}(z_{ij} - t_{ij}) = w^\sigma(S_M, V\backslash S_M)$ where S_M is the set of mediators defined by $\{m_i\}_{i\in V}$. Finally, the objective function (2) minimizes the imbalance defined by (1). The first term penalizes negative edges (i,j) connecting vertices in a same cluster (i.e., such that $x_{ij} = 0$) and the second term penalizes positive edges (i,j) connecting non-mediator vertices in different clusters (i.e, such that $x_{ij} = 1$ and $z_{ij} = 0$).

In Section 6 the performance of this formulation is compared with the ones of the two enumeration algorithms presented in the next section.

5 Enumeration algorithms

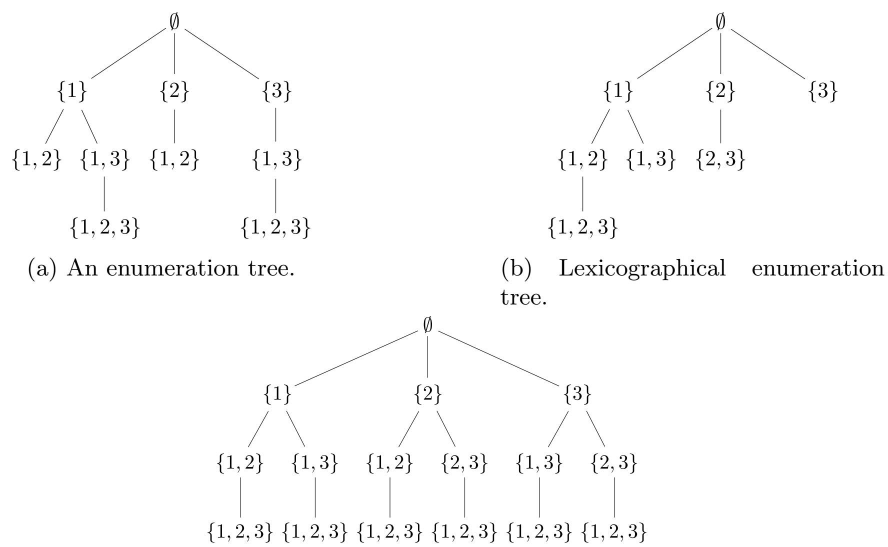
In this section, we present an alternative to the ILP based branch-and-bound algorithm, called *enumeration algorithms* for the optimal resolution of CCM. We first formally define the notion of enumeration algorithm (Section 5.1). Then, we study three simple enumeration strategies (called *policies*) and show that only one of them leads to an exhaustive enumeration of the mediator sets (Section 5.2). Finally, based on this policy, we propose two enumeration algorithms called A_1 and A_2 (Sections 5.3 and 5.4).

5.1 Enumeration tree and branching policy

Let an *enumeration tree* of a signed graph $G = (V, E, s)$ be a tree in which:

- each node is associated to a subset of V;
- the root corresponds to the empty set;
- each other node is associated to the set of its parent plus a new vertex.

Three enumeration trees are depicted in Figure 2.



(a) An enumeration tree.

(b) Lexicographical enumeration tree.

(c) Complete enumeration tree.

Figure 2: Three enumeration trees for $|V| = 3$.

An *enumeration algorithm* for CCM generates an enumeration tree in order to identify mediator sets of G. Solutions of the problem are then obtained by evaluating mediator sets identified. The evaluation of a set S_M consists in solving the CC problem associated with the signed graph induced by $V \backslash S_M$.

One of the main components of an enumeration algorithm is its *branching policy* $\pi : \mathcal{P}(V) \times V \mapsto \{true, false\}$ which indicates when a node should be created or not in the enumeration tree. More specifically, if S is a subset of V and i is a vertex in $V \backslash S$ then $\pi(S, i)$ returns $true$ if node $S \cup \{i\}$ must be created as a child of node S and $false$ otherwise. As a consequence, the size of the tree generated by an algorithm directly depends on its policy. If the branching policy always returns true ($\pi(S,i) = true$, $\forall S \in \mathcal{P}(V)$, $\forall i \in V\backslash S$), a complete tree of $\mathcal{O}(n!)$ nodes is created (see Figure 2c). Enumerating the sets in lexicographical order corresponds to the branching policy $\pi(S,i) =$ "$i > argmax_{s\in S}\ s$" (see example in Figure 2b). This policy leads to a smaller tree size by avoiding any repetition (i.e., each set is associated to no more than one node). However, the size of the corresponding tree ($2^{|V|}$) remains prohibitive and better alternatives are required to efficiently solve CCM.

5.2 Simple branching policies

An enumeration algorithm and its branching policy are said to be *exact* if they necessarily returns an optimal solution of CCM when the resolution time is not limited.

We first study three branching policies called $\pi_{\alpha\beta}$, π_α and π_β and show that only π_α is exact. Policy $\pi_{\alpha\beta}$ is an intuitive branching policy which generates a node only if it corresponds to a mediator set: $\pi_{\alpha\beta}(S,i) =$ "$S \cup \{i\}$ is a mediator set". Policies π_α and π_β are less restrictive and, thus, lead to larger enumeration trees:

- $\pi_\alpha(S,i) =$ "$S \cup \{i\}$ is α-feasible";
- $\pi_\beta(S,i) =$ "$S \cup \{i\}$ is β-feasible".

To determine if $\pi_{\alpha,\beta}$ is exact, we consider the following definition.

Definition 4. *(Björner and Ziegler (1992)) Let $\mathcal{F} \subseteq 2^S$ be a family of subsets of a set S. The tuple $(S,\mathcal{F})$ is an accessible system if and only if:*

(i) $\emptyset \in \mathcal{F}$,

(ii) if $X \in \mathcal{F}$ and $X \neq \emptyset$ then $\exists\ x \in X$ such that $X\backslash\{x\} \in \mathcal{F}$.

Let $\mathcal{M}$ be the family of all the mediator sets of a signed graph $G = (V,E,s)$. Similarly, let $\mathcal{A}$ and $\mathcal{B}$ be the family of all the α-feasible and β-feasible sets of G, respectively. The three following lemmas prove that the exactitude of the branching policies $\pi_{\alpha\beta}$, π_α and π_β depends on the fact that $(V,\mathcal{M})$, $(V,\mathcal{A})$ and $(V,\mathcal{B})$ are accessible systems or not. The proof of these lemmas are similar and only the first is provided.

Lemma 2. *$\pi_{\alpha\beta}$ is exact if and only if $(V,\mathcal{M})$ is an accessible system.*

Proof. Let S be a mediator set. If $(V,\mathcal{M})$ is an accessible system, there exists an ordering $(s_1, s_2, ..., s_{|S|})$ of the vertices in S such that $S\backslash\{s_1, s_2, ..., s_i\}$ is a mediator set for all $i \in \{1,2,...,|S|\}$. As a consequence, S can be reached by $\pi_{\alpha\beta}$ through the following branch: $\emptyset$, $\{s_{|S|}\}$, $\{s_{|S|}, s_{|S|-1}\}$, ..., S.

We now assume that $\pi_{\alpha\beta}$ does not enumerate all the mediator sets. Let S be a minimal mediator set which is not enumerated by $\pi_{\alpha\beta}$. Since all the mediator sets of size $|S|-1$ are enumerated by $\pi_{\alpha\beta}$, we deduce that $S\backslash\{s\}$ is not a mediator set, for all $s \in S$. Consequently, $(\mathcal{M},V)$ is not an accessible system. □

Lemma 3. *π_α is exact if and only if $(V, \mathcal{A})$ is an accessible system.*

Lemma 4. *π_β is exact if and only if $(V, \mathcal{B})$ is an accessible system.*

As summarized in Table 1, we characterize in the remaining of this section when $(V, \mathcal{M})$, $(V, \mathcal{A})$ and $(V, \mathcal{B})$ are accessible systems or are even matroids.

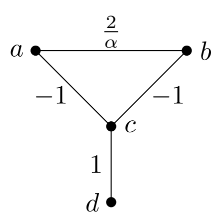
Tuple	$\alpha > 0$ $\beta \geqslant 0$	$\alpha = 0$ $\beta > 0$	$\alpha = 0$ $\beta = 0$
$(V, \mathcal{M})$	Non-accessible (Lemma 5)	Accessible (Lemma 7)	Matroid (Lemma 8)
$(V, \mathcal{A})$	Accessible (Lemma 10)	Matroid (Lemma 9)	
$(V, \mathcal{B})$	Non-accessible (Lemma 11)		

Table 1: Properties satisfied by $(V, \mathcal{M})$, $(V, \mathcal{A})$ and $(V, \mathcal{B})$.

Unfortunately, $\pi_{\alpha\beta}$ is not exact in the general case.

Lemma 5. *If $\alpha \neq 0$, then $(V, \mathcal{M})$ is not an accessible system.*

Proof. In the graph represented in Figure 3, $\{a, b, c\}$ is a mediator set but none of the subsets $\{a, b\}$, $\{a, c\}$ and $\{b, c\}$ is. □



(a) A signed graph for which $\{a, b, c\}$ is a mediator set.

Set	α**-feasible**?	β**-feasible**?
$\{a, b, c\}$	yes $2 \leq \frac{2}{\alpha}\alpha$	yes $0 \leq \beta$
$\{a, b\}$	yes $0 \leq \frac{2}{\alpha}\alpha$	no $2 > 0\beta$
$\{a, c\}$ or $\{b, c\}$	no $1 > 0\alpha$	if $\beta \neq 0$ $1 \leq (\frac{2}{\alpha} + 1)\beta$

(b) Table which shows that for each $i \in \{a, b, c\}$, $\{a, b, c\} \setminus \{i\}$ is not a mediator set.

Figure 3: Example which proves that $(V, \mathcal{M})$ is not an accessible system when $\alpha \neq 0$.

To prove that $\pi_{\alpha\beta}$ is exact when $\alpha = 0$, we first consider the following lemma.

Lemma 6. *If S is a mediator set and $\alpha \leqslant \beta$, then $\exists s \in S$ such that $S \setminus \{s\}$ is β-feasible.*

Proof. Let assume that for each $s \in S$, $S \setminus \{s\}$ is not β-feasible. Hence, the following inequality holds for all $s \in S$

$$\beta w^+(S \setminus \{s\}, V \setminus \{S \setminus \{s\}\}) < w^-(S \setminus \{s\}, V \setminus \{S \setminus \{s\}\})$$

or, equivalently

$$\beta w^+(S \setminus \{s\}, V \setminus S) + \beta w^+(s, S) < w^-(S \setminus \{s\}, V \setminus S) + \beta w^-(s, S)$$

By summing up this inequality for each $s \in S$ we obtain

$$(|S| - 1)\beta w^+(S, V \setminus S) + \beta \underbrace{\sum_{s \in S} w^+(s, S)}_{=2w^+(S)} < (|S| - 1)w^-(S, V \setminus S) + \underbrace{\sum_{s \in S} w^-(s, S)}_{=2w^-(S)}$$

Since S is β-feasible, $(|S| - 1)\beta w^+(S, V \setminus S) \geqslant (|S| - 1)w^-(S, V \setminus S)$, which together with the previous inequality leads to

$$\beta w^+(S) < w^-(S)$$

Since $\alpha \leqslant \beta$ this last inequality contradict the α-feasibility of S. □

We now prove that $(V, \mathcal{M})$ is an accessible system when $\alpha = 0$.

Lemma 7. *If $\alpha = 0$, then $(V, \mathcal{M})$ is an accessible system.*

Proof. If $\alpha = 0$, the weight of each edge in a mediator set S_M is non-negative. Hence, any subset of S_M is α-feasible. We deduce from Lemma 6 that there exists at least one vertex $s \in S_M$ such that $S_M \setminus \{s\}$ is additionally β-feasible. □

Note, that when $\alpha = \beta = 0$, $(V, \mathcal{M})$ is not only an accessible system but also a *matroid.*

Lemma 8. *If $\alpha = \beta = 0$, then $(V, \mathcal{M})$ is a matroid.*

Proof. Since $\alpha = \beta = 0$ the weight of each edge in S_M and between S_M and $V \setminus S_M$ is necessarily non-negative. This applies to any subset of S_M. □

The following lemma can be proved similarly.

Lemma 9. *If $\alpha = 0$, then $(V, \mathcal{A})$ is a matroid.*

Lemma 8 ensures that, when both α and β are null, $\pi_{\alpha\beta}$ is exact. However, in this case, an enumeration algorithm based on this policy is not the best approach to solve CCM. Indeed, when $\alpha = \beta = 0$, an optimal solution of CCM can be obtained by identifying the maximal mediator set S_M and solving CC on the remaining vertices $V \setminus S_M$ (Figueiredo and Moura, 2013). The set S_M can easily be identified as it contains all the vertices with adjacent edges with only non-negative weights.

Since $\pi_{\alpha\beta}$ is not exact in the general case, we now focus on π_α et π_β. The two next lemmas show that π_α is exact and that π_β is not.

Lemma 10. *For any $\alpha > 0$ $(V, \mathcal{A})$ is an accessible system.*

Proof. Let assume that, for each vertex s of an α-feasible set S, $S \setminus \{s\}$ is not α-feasible:

$$\alpha w^+ (S \setminus \{s\}) - w^- (S \setminus \{s\}) < 0 \quad \forall s \in S. \tag{15}$$

Summing up these inequalities for each $s \in S$, we obtain

$$(|S| - 2) \left(\alpha w^+ (S) - w^- (S)\right) < 0, \tag{16}$$

since each edge (i, j), with $i, j \in S$, appears in each inequality (15) except when s is equal to i or j.

Equation (16) contradicts the α-feasibility of S □

Lemma 11. *For any $\beta \geqslant 0$ $(V, \mathcal{B})$ is not an accessible system.*

Proof. Consider a graph composed of two vertices linked by an edge of weight -1. The set $\{s, t\}$ is β-feasible while $\{s\}$ and $\{t\}$ are not.

□

Our two enumeration algorithms A_1 and A_2 are based on π_α since, as summarized in Table 1, π_β and $\pi_{\alpha\beta}$ are not exact in most of the cases.

5.3 Algorithm A_1

In this section, we present our first enumeration algorithm A_1 and its branching policy π_{A_1}.

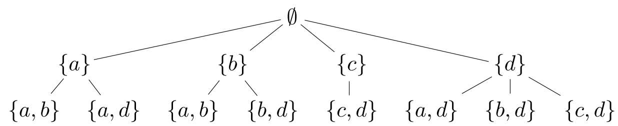
The enumeration and the evaluation of the mediator sets are two time consuming steps of an enumeration algorithm. Consequently, we introduce in Section 5.3.1 an exact branching policy π_{A_1} which produces trees significantly smaller than π_α. Moreover, to speed up the evaluation step, we prove in Section 5.3.2 that only maximal mediator sets need to be evaluated.

5.3.1 Branching policy π_{A1}

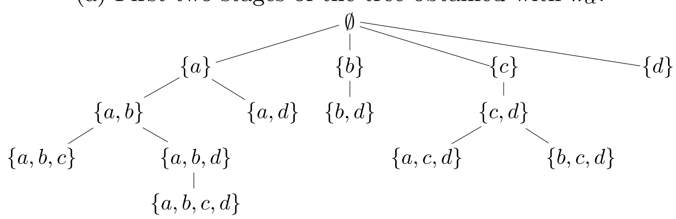
Lemmas (5) to (11) prove that π_α is exact while π_β and $\pi_{\alpha\beta}$ are not. Unfortunately, the enumeration tree generated by π_α may be huge (even larger than the lexicographical order policy) since π_α does not avoid repetitions (i.e., several nodes of the generated tree may correspond to the same set). This is exampliﬁed by the enumeration tree represented in Figure 4a in which all the sets of size 2 are duplicated.

It would be tempting to combine π_α with the lexicographical policy and only enumerate in lexicographical order the sets which are α-feasible. However, this policy would not be exact. Indeed, if in Figure 2b the set $\{1, 2\}$ is not α-feasible, then the set $\{1, 2, 3\}$ can not be generated even if it is a mediator set.

The following lemma enables to design an exact branching policy without repetitions.



(a) First two stages of the tree obtained with π_α.



(b) Tree obtained with π_{A_1}.

Figure 4: Enumeration trees obtained for different policies for the graph presented in Figure 3a.

Lemma 12. *If $S \subset V$ is α-feasible and $v \in argmin_{i \in S}\ \alpha w^+(i,S) - w^-(i,S)$, then $S \backslash \{v\}$ is α-feasible.*

Proof. Lemma 10 ensures that there exists $k \in S$ such that $S\backslash\{k\}$ is α-feasible:

$$\alpha w^+(S) - w^-(S) - (\alpha w^+(k,S) - w^-(k,S)) \geq 0. \tag{17}$$

Let us assume that there exists a vertex $v \in argmin_{i\in S}\ \alpha w^+(i,S) - w^-(i,S)$ such that set $S\backslash\{v\}$ is not α-feasible:

$$\alpha w^+(S) - w^-(S) - (\alpha w^+(v,S) - w^-(v,S)) < 0. \tag{18}$$

However, by definition of v we have

$$\alpha w^+(k,S) - w^-(k,S) \geq \alpha w^+(v,S) - w^-(v,S) \tag{19}$$

thus, Equations (17) and (18) can not both hold.

□

Let S be an α-feasible set. Lemma 12 ensures that by successively removing from S a vertex which minimizes $\alpha w^+(i,S) - w^-(i,S)$ (i.e., a vertex of S which contribution to the α-feasibility of S is minimal), a serie of α-feasible sets is obtained. In other words, S can be reached by a branching policy which uses this condition.

We can now exhibit the complete expression of π_{A_1}. Branching policy $\pi_{A_1}(S,i)$ returns true if and only if:

- $S \cup \{i\}$ is α-feasible; and
- $i = \min\ argmin_{s\in S}\ (\alpha w^+(s,S) - w^-(s,S))$.

A minimization is used in the second condition to ensure that there are no repetitions in the enumeration tree whenever several vertices in S have a minimal contribution to the α-feasibility of S.

We now present how the evaluation step of an enumeration algorithm can be improved.

5.3.2 Evaluation of the generated mediator sets

In order to solve the CCM problem, an enumeration algorithm must evaluate the mediator sets it generates. The evaluation of a mediator set S_M consists in solving the CC problem on a graph composed of the vertices $V \backslash S_M$. This step can be performed after the enumeration of the mediator sets or in parallel.

For a given set $S \subseteq V$, let P^S be an optimal partition of the CC problem over $V \backslash S$. Since CC is NP-hard, reducing the number of evaluated sets could have significant impact on the resolution time of an enumeration algorithm. The next lemma ensures that we can only evaluate maximal mediator sets.

Lemma 13. *Adding a vertex to the mediator set can not deteriorate the optimal value of CCM.*

Proof. Let S be a set of mediators and s a vertex in $V \backslash S$. We show that the optimal value of the CC problem over $V \backslash S$ (i.e., $I(P^S)$) is greater than or equal to the one of the CC problem over $V \backslash (S \cup \{s\})$ (i.e., $I(P^{S \cup \{s\}})$).

Let $P^S = \{S_1, ..., S_k\}$ and assume without loss of generality that $s \in S_1$. According to Equation (1),

$$I(\{S_1 \setminus \{s\}, \ldots, S_k\}) \;=\; I(P^S) \;-\; w^-(\{s\}, S_1) \;-\; \sum_{2 \leq j \leq k} w^+(\{s\}, S_j). \quad (20)$$

We can then conclude that

$$I(P^S) \geq I(\{S_1 \setminus \{s\}, \ldots, S_k\}) \geq I(P^{S \cup \{s\}}). \quad (21)$$

□

Corollary 1. *Let $S, S' \subseteq V$ be two mediator sets in G such that $S \subseteq S'$. Then $I(P^S) \geq I(P^{S'})$.*

Consequently, we only test maximal mediator sets in our algorithms.

5.3.3 Pseudo-code of Algorithm A_1

To solve CCM, Algorithm A_1 generates all the mediator sets by calling the recursive function $A1Enumeration(G, \emptyset)$ (see Algorithm 1) and returning one which minimizes the imbalance. Lines 2 and 3 of function $A1Enumeration$ enable to generate all the child nodes of node S which satisfy branching policy π_{A_1}. The mediator sets are evaluated on Line 6 if no mediator set is found in the subtree (i.e., if $L = \emptyset$). Note that this does not prevent A_1 from evaluating non maximal mediator sets.

Algorithm 1: Recursive function $A1Enumeration$.

Data: $G = (V, E, s)$: weighted signed undirected graph
$S \subset V$

Result: L : list of mediator sets which strictly include S and their associated optimal value of the CC problem

$L \leftarrow \emptyset$
for $i \in V \backslash S$ **do**
  if $\pi_{A_1}(S, i)$ **then**
    $L \leftarrow L \cup A1Enumeration(G, S \cup \{i\})$
if $L = \emptyset$ *and* S *is* β*-feasible* **then**
  $v^* \leftarrow$ optimal value of the CC problem of $V \backslash S$
  $L \leftarrow L \cup \{(S, v^*)\}$
return L

Lemma 14. *Algorithm A_1 may evaluate non maximal mediator sets.*

Proof. Figure 5b represents the enumeration tree obtained using policy π_{A_1} over the graph represented in Figure 5a.

(a) (b)

Figure 5: (a) A graph and (b) its corresponding enumeration tree obtained with Algorithm A_1.

Since $\{b\}$ is a mediator set and a leaf of the tree, it will necessarily be evaluated during Algorithm A_1. However, it is not a maximal mediator set as it is included in $\{a, b\}$. □

Algorithm A_1 enumerates exhaustively the mediator sets. We now define a second exact enumeration algorithm called A_2 which leverage linear relaxations to significantly reduce the size of its enumeration tree.

5.4 Algorithm A_2

Algorithm A_2 is based on the recursive function $A2Enumeration$, represented in Algorithm 2, which is slightly different from $A1Enumeration$ in order to reduce the size of the enumeration tree. An upper bound UB which corresponds to the imbalance of a known feasible solution of the CCM problem is given as an input. Moreover, at each node S, we compute the value v_r of the linear relaxation of

CCM in which the vertices in S are imposed to be included in the mediator set (Line 2). If v_r is greater than UB, this sub-tree can not lead to a better solution and it is pruned. Finally, UB is updated whenever a better integer solution is obtained (Line 10).

Algorithm 2: Recursive function *A2Enumeration*

Data: $G = (V, E, s)$: weighted signed undirected graph
$S \subset V$
UB: best known upper bound of CCM (global variable)

Result: L: list of all mediator sets which strictly include S and their associated optimal value of the CC problem

1 $L \leftarrow \emptyset$
2 $v_r \leftarrow$ optimal value of the linear relaxation of the CCM problem in which S is forced to be included in the mediator set
3 **if** $v_r < UB$ **then**
4   **for** $i \in V \backslash S$ **do**
5   **if** $\pi_{A_1}(S, i)$ **then**
6             $L \leftarrow L \cup A2Enumeration(G, S \cup \{i\})$
7   **if** $L = \emptyset$ *and* S *is* β*-feasible* **then**
8         $v^* \leftarrow$ optimal value of the CC problem of $V \backslash S$
9         $L \leftarrow L \cup \{(S, v^*)\}$
10         $UB = \min(UB, v^*)$
11 **return** L

To provide an initial upper bound, we use the greedy heuristic represented in Algorithm 3. This heuristic tries to find a list of mediator sets L such that each vertex in V appears in at least one of them. For this purpose the list *notInASet* initially contains all the vertices (Line 2) and each time a vertex is added to a mediator set, it is removed from this list (Line 6 and 10). Each pass of the while loop Line 3 tries to create a mediator set S_M starting with a candidate vertex from *notInASet* (Line 4 and 5). Vertices are then added to S_M by successively selecting vertices which improve the most the α and the β-feasibilities of S_M (Line 7 and 11). Prior to adding S_M to L, we test if S_M is a mediator set (Line 12). Indeed, if a candidate vertex is not included in any mediator set of size 2, S_M can not be a mediator set. In that case, the greedy algorithm may not return any mediator set which includes this vertex.

Algorithm 3: Greedy heuristic for the CCM problem H_G.

Data: $G = (V, E, s)$: weighted signed undirected graph
Result: L: list of mediator sets

$L \leftarrow \emptyset$
$notInASet \leftarrow V$ // List of vertices which does not appear in any mediator set found
while $notInASet \neq \emptyset$ **do**
  $candidate \leftarrow notInASet[1]$
  $S_M \leftarrow \{candidate\}$
  $notInASet \leftarrow notInASet \backslash \{candidate\}$
  $v \leftarrow argmax_{i \in V \backslash S_M} \min(\alpha w^+(i, S_M) - w^-(i, S_M), \beta w^+(i, V\backslash S_M) - w^-(i, V\backslash S_M))$
  while $S_M \cup \{v\}$ *is a mediator set* **do**
    $S_M \leftarrow S_M \cup \{v\}$
    $notInASet \leftarrow notInASet \backslash \{v\}$
    $v \leftarrow argmax_{i \in V \backslash S_M} \min(\alpha w^+(i, S_M) - w^-(i, S_M), \beta w^+(i, V\backslash S_M) - w^-(i, V\backslash S_M))$
  if S_M *is a mediator set* **then**
    $L \leftarrow L \cup S_M$
return L

Algorithm A_2 starts by calling the greedy heuristic. Each maximal mediator set returned is then evaluated and the best imbalance obtained constitutes the initial upper bound UB. The exact enumeration is then performed by calling $A2Enumeration(G, \emptyset, UB)$.

5.5 Implementation improvements

To improve the efficiency of A_1 and A_2, several implementation choices have been made.

At each node, the α and the β-feasibility are not computed from scratch. They are instead deduced from the values obtained at the parent node. For example, let us consider a node $S \cup \{i\}$ son of node S. At node $S \cup \{i\}$, the α-feasibility of node S has already been tested. The value $\alpha w^+(S) - w^-(S)$ is thus known. We leverage this value to test the α-feasibility of node $S \cup \{i\}$ thanks to the equation:

$$\alpha w^+(S \cup \{i\}) - w^-(S \cup \{i\}) = \alpha w^+(S) - w^-(S) + \alpha w^+(i, S) - w^-(i, S). \quad (22)$$

Consequently, at each node $S \cup \{i\}$, we only compute the value $\alpha w^+(i, S) - w^-(i, S)$. A similar reasoning is considered for the β-feasibility tests.

Enumeration algorithms must both enumerate and evaluate mediator sets. The evaluation of a set S requires to solve a NP-hard problem and we know that it is not necessary if S is not a maximal mediator set. Consequently, it is not efficient to evaluate a set as soon as it is enumerated. An alternative would be to first enumerate all the mediator sets and then evaluate the one which are maximal. This approach has two drawbacks:

- when the resolution time is limited, enumerating all the mediator sets may not leave enough time to evaluate many mediator sets, leading to a solution of poor quality. In hard instances it can even lead to no solution at all;
- in A_2 evaluating mediator sets may enable to improve the upper bound UB, thus reducing the size of the enumeration tree. If the mediator sets are evaluated after the enumeration, this bound can not be strengthened during the enumeration.

Consequently, our algorithms alternate between the enumeration and the evaluation steps until the algorithm or the time is over.

6 Computational experiments

We compare the performances of A_1, A_2 and the formulation presented in Section 4 on two datasets composed of random instances (Section 6.1) and instances obtained from the vote of the members of the european parliament (Section 6.2)[1]. We use a 3.50GHz Intel(R) Xeon(R) CPU E31280 equipped with 31GByte of RAM and the linear programs are solved with CPLEX 12.9.

For each instance I considered, let $\bar{\alpha}_I = \dfrac{\sum_{(i,j)\in E^-} w_{ij}}{\sum_{(i,j)\in E^+} w_{ij}}$ be the lowest value of α for which V is a mediator set. The solution in which V is the mediator set is always optimal since it leads to an imbalance of 0. Consequently, the problem is trivial for any value $\alpha \geq \bar{\alpha}_I$. To evaluate our methods over non-trivial problems, we consider for each instance I the three following values of α: $0.25\,\bar{\alpha}_I$, $0.5\,\bar{\alpha}_I$ and $0.75\,\bar{\alpha}_I$.

6.1 Random dataset

We randomly generate instances with 30 to 50 vertices and with densities 0.2, 0.5 and 0.8 using the erdos.renyi.game function from R's "igraph" library (see (Csardi and Nepusz, 2006)). The density $\rho \in [0, 1]$ corresponds to the probability that an edge exists. The weight of the edges are uniformly generated in $[-1, 1]$.

For a given instance, let x^I be the value of the best solution returned by a method and let x^{LB} be the lower bound it provides. We define the *relative gap* as $100 \times \frac{|x^I - x^{LB}|}{x^I}$. Since A_1 and A_2 do not provide a lower bound, the lower bound obtained with CPLEX is used to compute their relative gap.

[1] datas are available via
`https://sites.google.com/site/zacharieales/2021-01_data_mediators.tar.gz?attredirects=0&d=1`

The execution time, the number of nodes generated and the relative gap of each method are presented in Table 2. Each entry corresponds to a mean value over 5 instances. The time limit of each method is fixed to 4 hours. The two first columns of Table 2 represent the size and density of the graphs. The next column contains the percentage of $\bar{\alpha}_I$ considered.

The resolution of our formulation through CPLEX appears to provide the best results on most of the instances. Algorithm A_2 is often close to CPLEX and is even able to beat it in 13% of the instances. CPLEX is known for the efficiency of its presolve algorithm which often enables to drastically reduce the size of a MILP and its fine-tuned heuristics which determine in particular on which variable to branch and which node to evaluate next. We posit that the efficiency of CPLEX over A_1 and A_2 is mainly due to these features which enable to optimally solve the problems with a significantly smaller number of nodes.

The differences in terms of resolution time and size of the enumerated trees between A_1 and A_2 highlight the efficiency of A_2 pruning mechanism.

We observe that the resolution times tend to increase with size of the graph, its density and $\bar{\alpha}_I$. This is not surprising as all these parameters are related to the complexity of the problem. The size of the graph determines the number of variables in the formulation and the number of branches to consider in the enumeration algorithms. The greater the density, the more complex the objective function. Finally, $\bar{\alpha}_I$ directly impacts the number of feasible solutions.

Most of the instances where A_2 beats CPLEX correspond to $\bar{\alpha}_I = 0.25$. This is due to the fact that the size of the maximal mediator sets decreases when α decreases, thus reducing the depth of the branches of the enumeration algorithms.

6.2 European parliament dataset

We now consider real world instances obtained from votes casted during the 7^{th} term of the european parliament from 2009 to 2014. The roll-call votes of all members of the european parliament (MEP) for all plenary sessions in this period are available on the website *It's Your Parliament* (Buhl & Rasmussen (2020)).

In order to obtain challenging instances, we selected countries with more than 30 MEP and three of the most controversial policy domains: agriculture, gender equality and economic. For each country 1 graph is generated for each domain. Similarly to Arinik et al. (2017), to each MEP is associated a vertex and the weight of an edge in $[-1, 1]$ represents the voting similarity between two MEP.

The results obtained for this dataset are presented in Table 3. Each value in this table corresponds to an average over three instances (one for each policy domain considered). The resolution time of CPLEX quickly increases with the size of the graphs and it is only able to provide feasible solutions for the three

$\lvert V\rvert$	ρ	$\bar{\alpha}_I\%$	**CPLEX**			A_2			A_1		
			Time	Gap	Nodes	Time	Gap	Nodes	Time	Gap	Nodes
30	0.2	25	**6s**	0%	127	79s	0%	9307	1082s	0%	13518449
		50	**5s**	0%	100	4011s	0%	504477	4581s	0%	51533226
		75	**4s**	0%	11	1459s	0%	243957	15480s	-	38968391
30	0.5	25	19s	0%	21	**8s**	0%	47	83s	0%	989595
		50	**41s**	0%	449	226s	0%	9013	1743s	0%	19697051
		75	**43s**	0%	1075	14400s	100%	732676	12460s	80%	116702495
30	0.8	25	48s	0%	232	**29s**	0%	31	55s	0%	134642
		50	162s	0%	2263	**58s**	0%	343	564s	0%	6111397
		75	**519s**	0%	6991	14400s	62%	169606	11439s	20%	110705306
40	0.2	25	35s	0%	50	**33s**	0%	661	14400s	-	105226770
		50	**2628s**	0%	3323	13323s	2%	336233	14400s	3%	96866702
		75	**20s**	0%	117	5760s	0%	93792	6632s	0%	24934814
40	0.5	25	**1042s**	0%	1574	1058s	0%	46	5880s	0%	32044875
		50	**3118s**	0%	5557	5015s	3%	17471	14400s	-	93735047
		75	**8661s**	20%	28870	14400s	43%	61975	14400s	27%	83403944
40	0.8	25	**6774s**	0%	6273	7479s	0%	41	6936s	21%	2340239
		50	14400s	0%	64626	**12041s**	74%	183	14400s	91%	61945117
		75	14400s	**2%**	48616	14400s	22%	46455	14400s	56%	82982893
50	0.2	25	**709s**	0%	333	1123s	0%	4687	14400s	-	89114438
		50	13578s	-	6665	13002s	**42%**	118864	14400s	81%	81664916
		75	**286s**	0%	493	5761s	40%	42555	229519s	-	46021614
50	0.5	25	14400s	**0%**	3762	14400s	94%	0	14400s	-	96815443
		50	14400s	**0%**	9605	14400s	65%	3828	14400s	-	57217269
		75	14400s	100%	4355	14400s	86%	62194	14400s	**48%**	49755356
50	0.8	25	14400s	**0%**	5028	14400s	143%	0	14400s	182%	37134234
		50	14400s	**0%**	13984	14400s	94%	0	14400s	-	77940184
		75	14400s	101%	7463	14400s	**92%**	44826	14400s	98%	34718564

Table 2: Mean time in seconds, relative gap and number of enumerated nodes obtained for each method over the random instances. Each value is an average over five instances. On each line, the best result is in bold. A dash in the Gap column indicates that no solution is obtained for at least one of the instances.

n	**Country**	$\bar{\alpha}_I\%$	**CPLEX**			**A_2**		
			Time	Obj.	Nodes	Time	Obj.	Nodes
33	Romania	0.25	**10s**	0	53	**0**s	**0**	1
		0.5	**7**s	**0**	172	**0**s	**0**	1
		0.75	**7**s	**0**	172	**0**s	**0**	1
51	Poland	0.25	391s	0	30	**0**s	**0**	1
		0.5	1116s	0	658	**0**s	**0**	1
		0.75	149s	0	15	**0**s	**0**	1
59	Spain	0.25	2390s	0	669	**0**s	**0**	1
		0.5	2015s	0	76	**0**s	**0**	1
		0.75	614s	0	11	**0**s	**0**	1
72	UK	0.25	9977s	-	2	**9601**s	**1**	29388
		0.5	11006s	-	189	**9601**s	**1**	31937
		0.75	14407s	-	440	**4803**s	**0**	9827
87	France	0.25	14410s	-	5	**9601**s	**4**	16682
		0.5	14403s	-	29	**4803**s	**2**	6610
		0.75	14402s	-	19	**6**s	**0**	1
104	Germany	0.25	14403s	-	2	**9601s**	**0**	10215
		0.5	14403s	-	2	**7**s	**0**	1
		0.75	14403s	-	2	**17**s	**0**	1

Table 3: Mean time in seconds, objective value and number of enumerated nodes obtained on the instances from the european parliament. Each value is an average over three instances. On each line, the best result is in bold and a dash is used in column Obj. if no solution is obtained for at least one of the instances.

smallest instances. Algorithm A_2, however, is faster than CPLEX and always returns a solution. The efficiency of A_2 is partially due to its greedy heuristic which is very efficient on these real world instances. Indeed, it often returns a solution with no imbalance leading to an enumeration tree with only one node. This is not surprising as the instances are quite polarized along the lines of the political groups of the european parliament. However, the efficiency of A_2 is not only due to its greedy heuristic as the enumeration algorithm enables to improve the greedy solution in most instances with several nodes.

We conclude this section by highlighting advantages of the enumeration algorithms over the integer programming formulation. First, enumeration algorithms generate all the maximal mediator sets. A_1 even returns all the mediator sets. In the context of decision aid systems, providing a variety of relevant solutions is essential. The set of all the mediators could also be used to indicate the importance of each vertex v. For example, if only one vertex v is present in all the mediator sets, it indicates that it plays a major role in the graph. CPLEX can be tuned to generate a pool of solutions but it can not guarantee that all the maximal mediator sets or even all the optimal solutions are obtained. Secondly, the enumeration algorithms can easily be adapted to new definitions of mediator sets involving non-linear and non-convex constraints. The satisfaction of these constraints can be tested at the same time than the β-feasibility (Line 5 of Algorithm 1 and Line 7 of Algorithm 2).

7 Conclusions and perspectives

In this paper, we propose a new variant of the correlation clustering problem, called the correlation clustering problem with mediation, based on the work of Doreian and Mrvar (2009). After proving its NP-hardness we model it with an integer mathematical formulation. We also develop two enumeration algorithms A_1 and A_2 to solve optimally this problem and exhaustively enumerate all the maximal sets of mediators. These algorithms are based on properties of the mediator sets which enable to efficiently prune branches of the enumeration tree. Finally, we compare experimentally the performances of the formulation and of the enumeration algorithms on a dataset with random instances and on a second with real world instances obtained from european parliament votes. The resolution of the formulation with CPLEX gives better results on hard random instances but, unlike A_2 it fails to provide feasible solutions on the large real world instances.

A natural perspective to this work would be to improve the pruning technique of the enumeration algorithms by identifying additional properties of the mediator set to strengthen the branching policies. A new type of enumeration algorithm could also be introduced in which vertices are removed rather than added at each new node of the enumeration tree. Such algorithm could cut a branch as soon as a mediator set is reached. This approach could be particularly efficient when the maximal sets of mediators are large (i.e., for large values of parameters α and β). A last perspective would be to consider alternative definitions of a mediator set. The flexibility of the enumeration algorithms could allow the use of non-linear constraints. For some applications it could also be relevant to associate a label to each vertex (e.g., a political party) and to require that the proportion of each label in a mediator set is representative of its distribution in the graph.

References

Abramowitz AI, Saunders KL (2008) Is polarization a myth? The Journal of Politics 70(2):542–555

Agarwal G, Kempe D (2008) Modularity-maximizing network communities using mathematical programming. Eur Phys J B 66:409–418

Ahmadian S, Epasto A, Kumar R, Mahdian M (2019) Clustering without over-representation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, NY, USA, pp 267—-275

Ahmadian S, Epasto A, Kumar R, Mahdian M (2020) Fair correlation clustering. In: Chiappa S, Calandra R (eds) Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, PMLR, Online, Proceedings of Machine Learning Research, vol 108, pp 4195–4205

Ales Z, Knippel A, Pauchet A (2016) Polyhedral combinatorics of the k-partitioning problem with representative variables. Discrete Applied Mathematics 211:1–14

Aref S, Wilson MC (2019) Balance and frustration in signed networks. J Complex Networks 7(2):163–189

Arinik N, Figueiredo R, Labatut V (2017) Signed graph analysis for the interpretation of voting behavior. In: Proceedings of the International Conference on Knowledge Technologies and Data-Driven Business 2017 (i-Know 2017), Graz, Austria

Arinik N, Figueiredo R, Labatut V (2020) Multiplicity and Diversity: Analyzing the Optimal Solution Space of the Correlation Clustering Problem on Complete Signed Graphs. To appear in Journal of Complex Networks

Bansal N, Blum A, Chawla S (2004) Correlation clustering. Machine Learning 56:89–113

Bera S, Chakrabarty D, Flores N, Negahbani M (2019) Fair algorithms for clustering. In: Wallach H, Larochelle H, Beygelzimer A, Alché-Buc F, Fox E, Garnett R (eds) Advances in Neural Information Processing Systems, Curran Associates, Inc., vol 32, pp 4954–4965

Björner A, Ziegler GM (1992) Matroid applications, Cambridge University Press, chap 8, pp 284–357

Borgatti S (2003) The key player problem. In: Breiger R, Carley K, Pattison P (eds) Dynamic Social Network Modeling and Analysis, Workshop Summary & Papers, National Academy of Sciences Press, Washington-DC, pp 241–252

Borgatti SP (2006) Identifying sets of key players in a social network. Comput Math Organiz Theor 12(1):21–34

Brusco M, Steinley D (2009) Integer programas for one- and two-model blockmodeling based on prespecified image matrices for structural and regular equivalence. Journal of Mathematical Psychology 53:577–585

Buhl & Rasmussen (2020) It's your parliament. http://www.itsyourparliament.eu/

Cartwright D, Harary F (1956) Structural balance: A generalization of heiderŽs theory. Psychological Review 63:277–293

Chierichetti F, Kumar R, Lattanzi S, Vassilvitskii S (2017) Fair clustering through fairlets. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, NIPS'17, pp 5036—5044

Csardi G, Nepusz T (2006) The igraph software package for complex network research. InterJournal Complex Systems:1695

DasGupta B, Enciso G, Sontag E, Zhang Y (2007) Algorithmic and complexity results for decompositions of biological networks into monotone subsystems. Biosystems 9(1):161–178

Davis JA (1967) Clustering and structural balance in graphs. Human Relations 20(2):181–187

Demaine ED, Emanuel D, Fiat A, Immorlica N (2006) Correlation clustering in general weighted graphs. Theoretical Computer Science 361:172–187

Doreian P, Mrvar A (2009) Partitioning signed social networks. Social Networks 31:1–11

Figueiredo R, Frota Y (2014) The maximum balanced subgraph of a signed graph: Applications and solution approaches. European Journal of Operational Research 236(2):473 – 487

Figueiredo R, Moura G (2013) Mixed integer programming formulations for clustering problems related to structural balance. Social Networks 35(4):639–651

Figueiredo RMV, Labbé M, de Souza CC (2011) An exact approach to the problem of extracting an embedded network matrix. Computers & Operations Research 38(11):1483–1492

Gleich DF, Veldt N, Wirth A (2018) Correlation Clustering Generalized. In: Hsu WL, Lee DT, Liao CS (eds) 29th International Symposium on Algorithms and Computation (ISAAC 2018), Dagstuhl, Germany, Leibniz International Proceedings in Informatics (LIPIcs), vol 123, pp 44:1–44:13

Hansen P, Jaumard B (1997) Cluster analysis and mathematical programming. Mathematical Programming 79:191–215

Harary F (2002) Signed graphs for portfolio analysis in risk management. IMA Journal of Management Mathematics 13(3):201–210

Heider F (1946) Attitudes and cognitive organization. Journal of Psychology 21(1):107–112

Johnson EL, Mehrotra A, Nemhauser GL (1993) Min-cut clustering. Mathematical Programming 62:133–151

Kalhan S, Makarychev K, Zhou T (2019) Correlation clustering with local objectives. In: Wallach H, Larochelle H, Beygelzimer A, Alché-Buc F, Fox E, Garnett R (eds) Advances in Neural Information Processing Systems, Curran Associates, Inc., vol 32, pp 9346–9355

Kolluri R, Shewchuk JR, O'Brien JF (2004) Spectral surface reconstruction from noisy point clouds. In: Eurographics/ACM SIGGRAPH Symposium on Geometry processing, pp 11–21

Kropivnik S, Mrvar A (1996) An analysis of the slovene parliamentary parties network. Metodološki Zvezki / Advances in Methodology and Statistics 12:209–216

Levorato M, Figueiredo R, Frota Y, Drummond L (2017) Evaluating balancing on social networks through the efficient solution of Correlation Clustering problems. EURO Journal on Computational Optimization 5(4):467–498

Li CT, Lin SD, Shan MK (2011) Finding influential mediators in social networks. In: Proceedings of the 20th Int. Conf. Companion on World Wide Web, Association for Computing Machinery, New York, NY, USA, pp 75—76

Li P, Dau H, Puleo G, Milenkovic O (2017) Motif clustering and overlapping clustering for social network analysis. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp 1–9

Mehrotra A, Trick MA (1996) A column generation approach for graph coloring. INFORMS Journal of Computing 8:344–354

Mehrotra A, Trick MA (1998) Cliques and clustering: A combinatorial approach. Operations Research Letters 22(1):1 – 12

Ortiz-Arroyo D (2010) Discovering Sets of Key Players in Social Networks, Springer London, London, pp 27–47

Puleo GJ, Milenkovic O (2018) Correlation clustering and biclustering with locally bounded errors. IEEE Trans on Information Theory 64(6):4105–4119

Veldt N, Gleich D, Wirth A (2018) A correlation clustering framework for community detection. In: Proceedings of the 2018 World Wide Web Conference, Geneve, CHE, pp 439—-448