The K-partitioning problem: Formulations and branch-and-cut

Zacharie Alès, Arnaud Knippel

The K-Partitioning Problem: Formulations and Branch-and-Cut

Zacharie ALES [1]

UMA, CEDRIC, ENSTA Paris, Institut Polytechnique de Paris

Arnaud KNIPPEL [2]

LMI, INSA Rouen Normandie, France

Abstract

The K-partitioning problem consists in partitioning the nodes of a complete graph $G = (V, E)$ with weights on the edges in exactly K clusters such that the sum of the weights of the edges inside the clusters is minimized. For this problem, we propose two *node-cluster* formulations adapted from the literature on similar problems as well as two *edge-representative* formulations. We introduced the first edge-representative formulation in a previous work [4] while the second is obtained by adding an additional set of edge variables. We compare the structure of the polytopes of the two edge-representative formulations and identify a new family of facet-defining inequalities.

The quality of the linear relaxation and the resolution times of the four formulations are compared on various data sets. We provide bounds on the relaxation values of the node-cluster formulations which may account for their low performances. Finally, we propose a branch-and-cut strategy, based on the edge-representative formulations, which performs even better.

Keywords: graph partitioning, combinatorial optimization, integer programming, polyhedral approach, branch-and-cut, extended formulation

1 Introduction

In this paper we study a graph partitioning problem called the K-partitioning problem. Consider $G = (V, E)$ a complete graph with weights w_{ij} on each edge ij of E. The graph partitioning problem consists in partitioning the set $V = \{1, \ldots, n\}$ into non-empty subsets called *clusters* such that the sum

[1] Email: `zacharie.ales@ensta-paristech.fr`

[2] Email: `arnaud.knippel@insa-rouen.fr`

of the weights of the edges in the clusters is minimized. Variants of this problem have been extensively studied in the literature [8,24,26] and used in numerous applications such as processors load balancing [25,41] and image segmentation [14,32].

Grötschel and Wakabayashi [22,23] propose an integer programming formulation for a clustering problem in which the size and the number of clusters are not bounded. They study the polyhedral structure of the problem and present a cutting plane algorithm. Osten et al. [37] consider the clique partitioning problem in which each cluster must correspond to a clique. They introduce families of valid inequalities which are generalizations of the ones presented by Grötschel and Wakabayashi [23]. They also discuss lifting and patching techniques to define new facets.

Graph partitioning variants in which the sizes of the clusters are constrained have also been considered. Sorensen et al [39,40] study the *simple graph partitioning problem* in which a cluster may contain at most $b \in \mathbb{Z}^+$ nodes. They introduce a branch-and-cut algorithm. Several studies consider both upper and lower bounds on the number of nodes. Labbé and Öszoy [31] propose such a formulation and investigate the associated polytope. Johnson et al. [27] introduce a column generation approach.

Another variant of graph partitioning consists in constraining the number of clusters K. If $K = 2$, the problem is nothing but the *max-cut problem* [7]. Kaibel et al. [28] consider a formulation in which the number of clusters is less than a given bound q and propose a method to deal with its symmetry directly at the nodes of a branch-and-cut algorithm. Several papers also investigate the polyhedral structure of formulations with at most or at least K clusters [12,13,15].

Our present work is motivated by an application in dialogue analysis [1,5,6]. The aim of this application is to provide a decision aid software to help the identification of dialogical regularities. In order to guide the identification of such regularities, several dialogical patterns have to be clustered. Whenever the number of patterns is sufficiently low, an exact resolution of the associated clustering problem is possible and the obtained solutions have proved, through an expert evaluation, to be significantly more relevant than the ones obtained with various heuristics [1,5]. This software must provide relevant and user-friendly parameters. Setting the number of clusters satisfies both of these criteria as it corresponds to the number of dialogical regularities sought by the user. This is why we are considering the *K-partitioning problem* which consists in partitioning a complete graph into exactly K non-empty clusters. Nevertheless, some of our results naturally extend to the cases of at most K or at least K clusters (see end of Section 3).

Let n and m denote the number of nodes and edges in the graph, respectively. Few papers have considered the case in which the number of clusters is fixed without any additional constraint on the number of nodes allowed in each cluster. For example, the *equipartition problem* consists in partitioning the nodes into K clusters, each containing exactly $\frac{n}{K}$ nodes. Lisser and

Rendl [35] consider semidefinite and quadratic programming approaches while Mitchell [36] proposes a branch-and-cut algorithm for the problem. Whenever the weights of all the edges are negative Goldschmit and Hochbaum [21] prove that the problem can be solved in $O(n^{k^2/2-3k/2+4}\,T(n,m))$ time, where $T(n,m)$ is the time required to find the minimum (s,t)-cut on a graph with n nodes and m edges. However, this problem is known to be $\mathcal{NP}$-hard for the general case [20]. Most of the linear programming approaches in the literature for the exact resolution of the K-partitioning problem are based on *node-cluster* variables z_i^k taking value 1 if node i is assigned to cluster k. Unfortunately, these variables induce symmetry in linear programs even if additional constraints may partially alleviate it [9,16]. We introduced a symmetry-free formulation called *edge-representative formulation* (F_{er}) and we studied the associated polyhedron [3,4]. In this paper we show how this formulation can be extended to a new formulation (F_{ext}). Then, we compare the latter with (F_{er}) and two formulations with node-cluster variables adapted from the literature.

The next section is dedicated to the presentation of these four formulations. In Section 3 the polytopes (P_{ext}) and (P_{er}), associated with the linear relaxation of the edge-representative formulations, are compared. We introduce a new family of valid inequalities called *sub-representative inequalities*. We determine the conditions under which this family, as well as three families of constraints from (F_{ext}), are facet-defining of (P_{ext}). We also show that under the same conditions, these inequalities are facet-defining for two partitioning variants with at least K clusters or at most K clusters. A branch-and-cut algorithm, based on (F_{ext}) and a thorough cutting-plane step at its root node, is described in Section 4. Finally, in Section 5 we present numerical results and give bounds on the relaxation values of the node-cluster formulations.

2 Formulations

We first present the edge-representative formulation (F_{er}) introduced in [3,4]. We then show how it can be extended into a new formulation called (F_{ext}). We also adapt to our problem two formulations from the literature based on node-cluster variables.

2.1 Edge-representative formulation (F_{er})

Grötschel and Wakabayashi [22,23] consider the *clique partitioning problem* (CPP) which consists in partitioning a complete weighted graph into cliques so that the weight of the multicut is minimized. The authors restrict neither the number nor the size of the clusters. They introduce a formulation of the (CPP) based on edge variable x_{ij} for all ij in E. Variable x_{ij} takes value 1 if ij is inside a cluster of the partition and 0 otherwise. Note that x_{ij} and x_{ji} represent the same variable. To formulate the clique partitioning problem they consider the two following families of constraints:

$$x_{ij} \in \{0,1\} \quad \forall ij \in E, \tag{1}$$

$$x_{ij} + x_{ik} - x_{jk} \leq 1 \quad \forall i \in V \quad \forall j,k \in V \backslash \{i\},\ j < k. \tag{2}$$

Inequalities (2), called *triangle inequalities*, ensure that if two edges of a triangle are in the same cluster, then the third one is necessarily in the same cluster. Clearly the points in $\mathbb{R}^{\frac{n(n-1)}{2}}$ which satisfy (1) and (2) correspond exactly to the solutions of the clique partitioning problem.

Another way to formulate this problem consists in using node-cluster variables, which enables one to easily fix K but has the disadvantage of inducing a lot of symmetry. In [3,4] we considered Grötschel and Wakabayashi's edge formulation and presented a way of fixing the number of clusters to K, without adding any symmetry to the problem, by considering node variables r_v for all v in V (called *representative variables*). The idea of using such variables was first introduced in [11] for a node coloring problem. A slightly different idea has also been used in [9] for a node-cluster formulation of a difficult partitioning problem variant with an additional quadratic constraint. We adapt this idea in formulation (F_{nc2}) presented in Section 2.3. In the edge-representative formulation (F_{er}), variable r_v takes value 1 if v is the node with the lowest index of its cluster and 0 otherwise. Node v is said to be the *representative* of its cluster if r_v is equal to 1. Formulation (F_{er}) contains the following constraints:

$$0 \leq r_i \leq 1 \quad \forall i \in V \tag{3}$$

$$r_j + x_{ij} \leq 1 \quad \forall i,j \in V,\ i < j \tag{4}$$

$$r_j + \sum_{i=1}^{j-1} x_{ij} \geq 1 \quad \forall j \in V \tag{5}$$

$$\sum_{i=1}^{n} r_i = K. \tag{6}$$

Inequalities (4) ensure that node j cannot be a representative if it is not the lowest node of its cluster (*i.e.,* each cluster contains at most one representative). Inequalities (5) express the fact that j is a representative if nodes $1, \ldots, j$ are not in the same cluster (*i.e.,* each cluster contains at least one representative). The number of clusters is set to K thanks to Equation (6).

The *edge-representative formulation* (F_{er}) is obtained by considering Constraints (1)-(6)

$$(F_{er}) \begin{cases} \text{minimize} & \displaystyle\sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to } (1) - (6) \end{cases}.$$

Note that no integrality constraint is used for the representative variables. Constraints (1), (3), (4) and (5) ensure that they are either equal to one or zero.

The number of variables in (F_{er}) can be reduced to $\frac{n(n-1)}{2} + n - 3$ due to the following substitutions:

- $r_1 = 1$, since node 1 is always the lowest of its cluster;
- $r_2 = 1 - x_{1,2}$, since node 2 is always a representative except if it is in the same cluster as node 1;
- $r_3 = K - 2 + x_{1,2} - \sum_{i=4}^{n} r_i$, using Constraint (6).

In the remainder of the paper, (F_{er}) corresponds to the formulation in which r_1, r_2 and r_3 are substituted. However, to simplify the notations, all the variables are kept in the expression of the formulation.

2.2 *Extended edge-representative formulation* (F_{ext})

In (F_{er}) Inequalities (4) and (5) are used to fix the values of the representative variables. This can also be achieved through quadratic constraints

$$r_j + \sum_{i=1}^{j-1} r_i x_{ij} = 1 \quad \forall j \in V. \tag{7}$$

Thanks to (7), a node j is either a representative ($r_j = 1$) or it is in the same cluster as exactly one node, which is less than it, which is a representative ($r_i\ x_{ij} = 1$). We linearize these quadratic constraints by adding a new set of edge variables $\tilde{x}_{ij}$ for all $ij \in E$ such that $\tilde{x}_{ij}$ is equal to $r_i\ x_{ij}$. Constraints (7) can be replaced by

$$r_j + \sum_{i=1}^{j-1} \tilde{x}_{ij} = 1 \quad \forall j \in V, \tag{8}$$

$$0 \leq \tilde{x}_{ij} \leq 1 \quad \forall ij \in E, \tag{9}$$

$$\tilde{x}_{ij} \leq x_{ij} \quad \forall ij \in E, \tag{10}$$

$$\tilde{x}_{ij} \leq r_i \quad \forall ij \in E,\ i < j \tag{11}$$

and

$$x_{ij} + r_i - \tilde{x}_{ij} \leq 1 \quad \forall ij \in E,\ i < j. \tag{12}$$

Constraints (10), (11) and (12) ensure that $\tilde{x}_{ij}$ is equal to $r_i x_{ij}$.

The number of variables can again be reduced through substitutions:

- $r_1 = 1$, $r_2 = 1 - x_{1,2}$ and $r_3 = K - 2 + x_{1,2} - \sum_{i=4}^{n} r_i$ as in formulation (F_{er});
- $\tilde{x}_{1j} = x_{1j}$ for all $j \in \{2, \ldots, n\}$ since node 1 is always the representative of its cluster ;
- $\tilde{x}_{2,3} = 1 - r_3 - x_{1,3}$, using Constraint (8) for j equal to 3 ;
- $r_j = 1 - \sum_{i=1}^{j-1} \tilde{x}_{i,j}$ using Constraint (8) for all $j \in \{4, \ldots, n\}$.

These $2n$ variables are substituted in the formulation. Thus, an extended formulation of (F_{er}) with $n(n-2)$ variables is obtained. However, to simplify the notations, the formulation is written without the substitutions in the following:

$$(F_{ext}) \begin{cases} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to } (1)-(4), (6) \text{ and } (8)-(12) \end{cases}.$$

The representative variables still do not require additional constraints to ensure their integrality. The same applies to the edge variables $\tilde{x}_{ij}$. Note that Constraints (4) are not necessary in (F_{ext}) to formulate the K-partitioning prolem. However, as detailed in Section 3, they ensure that the linear relaxation of (F_{ext}) is at least as good as the one obtained with (F_{er}).

2.3 Node-cluster formulations

Given a maximal number of clusters $K_{max} \leq n$, the node-cluster formulations associates to each node $i \in V$ and each cluster $k \in \{1, 2, \ldots, K_{max}\}$ a variable z_i^k taking value 1 if node i is assigned to cluster k and 0 otherwise.

This type of formulation has been frequently considered for problems in which the number or the size of the clusters is constrained. Although a few formulations, based only on node-cluster variables, have been considered [9,16] they are usually combined with the edge variables x_{ij} [10,12,17,18,19,28]. In this last case a variant of the following formulation is considered:

$$(F_{nc}) \begin{cases} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to} & x_{ij} \in \{0,1\} \quad \forall ij \in E \quad (1) \\ & x_{ij} + z_i^k - z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \ldots, K_{max}\} \quad (13) \\ & x_{ij} - z_i^k + z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \ldots, K_{max}\} \quad (14) \\ & -x_{ij} + z_i^k + z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \ldots, K_{max}\} \quad (15) \\ & \sum_{k=1}^{K_{max}} z_i^k = 1 \quad \forall i \in V \quad (16) \\ & z_i^k = 0 \quad \forall k > i \quad i \in V \quad k \in \{1, \ldots, K_{max}\} \quad (17) \\ & z_i^k \in \{0,1\} \quad \forall i \in V \quad \forall k \in \{1, \ldots, K_{max}\} \quad (18) \end{cases}.$$

Relations (13), (14) and (15) have the same purpose as the triangle inequalities. Equations (16) ensure that each node $i \in V$ is in exactly one cluster. Equations (17) alleviate some of the symmetry by imposing that each node $i \in V$ is not in a cluster whose index is greater than i.

This formulation enables one to obtain a partition of minimal cost with at most K_{max} clusters. Note that when all weights w_{ij} are strictly positive, the number of clusters in an optimal solution is always K_{max}. Indeed, given a partition π with $K < K_{max}$ clusters, a better solution can be obtained by removing one node from any cluster of size at least 2 and using it to create a new cluster. When some of the weights are negative, partitions with less than K_{max} cluster may be obtained.

We now present two adaptations of (F_{nc}) which formulate the K-partitioning problem.

2.3.1 Formulation (F_{nc1})

A first way to obtain a formulation of the K-partitioning problem consists in fixing K_{max} to K [12,16,17,28] and imposing that each cluster is non-empty thanks to the following inequalities:

$$\sum_{i \in V} z_i^k \geq 1 \quad \forall k \in \{1, \ldots, K_{max}\}. \tag{19}$$

Thus, the first node-cluster formulation is

$$(F_{nc1}) \begin{cases} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to } (1) \text{ and } (13) - (19) \end{cases}.$$

2.3.2 Formulation (F_{nc2})

Equations (17) do not alleviate all the symmetry induced by the node-cluster variables. However, a second node-cluster formulation of the K-partitioning without any symmetry can be obtained by fixing K_{max} to n [16,28]. In that case, Bonami *et al.* [9] showed that the remaining symmetry can be removed using the following inequalities which ensure that a node j can only be in a cluster i if node i is also in it:

$$z_j^i \leq z_i^i \quad \forall j > i. \tag{20}$$

It is interesting to note that, thanks to Relations (17) and (20), several *node-cluster variables* have the same role as variables used in (F_{er}) and (F_{ext}):

- for all $i \in V$, r_i and z_i^i both take value one if and only if i is the lowest node of its cluster;
- for all $i < j$, variables z_j^i and $\tilde{x}_{ij}$ take value one if and only if j is represented by i.

Thus, the number of clusters can be fixed as follows:

$$\sum_{i \in V} z_i^i = K. \tag{21}$$

The obtained formulation is:

$$(F_{nc2}) \begin{cases} \text{minimize} \ \sum\limits_{ij \in E} w_{ij} x_{ij} \\ \text{subject to (1), (13) − (18), (20) and (21)} \end{cases}.$$

3 Polyhedral results

Let us denote by (P_{er}) and (P_{ext}) the convex hulls of all feasible integer solutions of (F_{er}) and (F_{ext}), respectively. In this section, we show that the linear relaxation value of (F_{ext}) is at least as good as the one of (F_{er}). Then, we determine the dimension of (P_{ext}) with respect to K. Finally, we describe the conditions under which a new family of inequalities, called the *sub-representative inequalities*, as well as Inequalities (10), (11) and (12), are facet-defining for (P_{ext}).

3.1 Notations

Let R_{er}, R_{ext} and R_{ext2} denote the convex hulls of all feasible solutions of the linear relaxations of (F_{er}), (F_{ext}) and (F_{ext}) without Constraints (4), respectively. When considering formulation (F_{ext}), the characteristic vector $x^\pi \in \mathbb{R}^{n(n-2)}$, associated with a K-partition π, is composed of $\frac{n(n-1)}{2}$ edge components followed by $\frac{n(n-3)}{2}$ components related to the $\tilde{x}_{ij}$ variables which have not been substituted, that is

$$(x^\pi)^T = (x_{1,2}, \ldots, x_{n-1,n}, \tilde{x}_{2,4}, \ldots, \tilde{x}_{n-1,n}).$$

The component of a vector $\alpha \in \mathbb{R}^{n(n-2)}$ related to a variable $\tilde{x}_{ij}$ will be denoted by $\tilde{\alpha}_{ij}$.

A variable which has been substituted in (F_{ext}) is called *artificial*. The artificial variables are $\tilde{x}_{2,3}$, r_i $(i \in \{1, ..., n\})$ and $\tilde{x}_{1j} (j \in \{1, ..., j\})$. In a vector $\alpha \in \mathbb{R}^{n(n-2)}$, no component is associated to an artificial variable. However, in the following, such components may be mentioned to ease the understanding.

When considering the edge-representative formulation, the characteristic vector $x^\pi \in \mathbb{R}^{\frac{n(n+1)}{2}-3}$ contains $n-3$ representative components (r_4 to r_n) followed by $\frac{n(n-1)}{2}$ edge components.

Let U and W be two distinct subsets of V and α a vector of $\mathbb{R}^{n(n-2)}$. The terms $\alpha(U)$ and $\alpha(U, W)$ refer to the expressions $\sum\limits_{u=1}^{|U|} \sum\limits_{u'=u+1}^{|U|} \alpha_{uu'}$ and $\sum\limits_{u \in U} \sum\limits_{w \in W} \alpha_{uw}$, respectively.

Transformations

Let P be a polytope which can either denote P_{ext} or one of its face. In order to characterize the dimension of P we identify the number of linearly independent hyperplanes $H = \{x \in \mathbb{R}^{n(n-2)} \mid \alpha^T x = \alpha_0\}$ including P. To

Term	Explanation
$\sum_{i\in R_1} \tilde{\alpha}_{r_1 i}$	$i \in R_1$ is represented by r_1 only in π_1
$\sum_{i\in R_1} \sum_{j\in C_1\backslash R_1} \alpha_{ij}$	$i \in R_1$ is in the same cluster as $j \in C_1\backslash R_1$ only in π_1
$\sum_{i\in R_1} \tilde{\alpha}_{r_2 i}$	$i \in R_1$ is represented by r_2 only in π_2
$\sum_{i\in R_1} \sum_{j\in C_2\backslash R_2} \alpha_{ij}$	$i \in R_1$ is in the same cluster as $j \in C_2\backslash R_2$ only in π_2.

Table 1
Terms of Equation (23).

obtain a relation between the coefficients of H, we compare two K-partitions $\pi_1 = \{C_1, C_2, C_3, \ldots, C_K\}$ and $\pi_2 = \{\{C_1\backslash R_1\}\cup R_2, \{C_2\backslash R_2\}\cup R_1, C_3, \ldots, C_K\}$ with $R_1 \subset C_1$, and $R_2 \subset C_2$. Note that R_1 or R_2 may be empty. If x^{π_1} and x^{π_2} are both in P, the following equality is satisfied

$$\alpha^T x^{\pi_1} = \alpha^T x^{\pi_2}. \tag{22}$$

Equation (22) can be simplified by removing the coefficients which appear on both sides of the equation. As a consequence, a relation between the coefficients of H can be obtained by identifying the coefficients which differ between x^{π_1} and x^{π_2}.

Let us assume that the representative nodes of the clusters in partitions π_1 and π_2 are the same. Let r_1 and r_2 be the representatives of C_1 and C_2 in these two partitions, respectively. In that case, the relation between the coefficients of H induced by π_1 and π_2 is:

$$\begin{aligned} &\textstyle\sum_{i\in R_1}(\tilde{\alpha}_{r_1 i} + \sum_{j\in C_1\backslash R_1} \alpha_{ij}) + \sum_{i\in R_2}(\tilde{\alpha}_{r_2 i} + \sum_{j\in C_2\backslash R_2} \alpha_{ij}) \\ &= \textstyle\sum_{i\in R_1}(\tilde{\alpha}_{r_2 i} + \sum_{j\in C_2\backslash R_2} \alpha_{ij}) + \sum_{i\in R_2}(\tilde{\alpha}_{r_1 i} + \sum_{j\in C_1\backslash R_1} \alpha_{ij}). \end{aligned} \tag{23}$$

Table 1 provides additional explanations of Equation (23). This table only mentions the equation terms related to R_1 since the terms related to R_2 are similarly obtained.

Let R be $R_1 \cup R_2$. To help efficiently identifying the relation associated to partitions π_1 and π_2, we introduce an operator $\mathcal{T}$ called *transformation*:

$$\mathcal{T} : \{C_1, C_2, R\} \mapsto \{(C_1\backslash R) \cup (R \cap C_2), (C_2\backslash R) \cup (R \cap C_1)\}.$$

This transformation is represented in Figure 1.

The relation obtained through a given transformation satisfies the two following properties:

- its left-hand side corresponds to the sum of all the coefficients associated with a variable equal to 1 before the transformation;

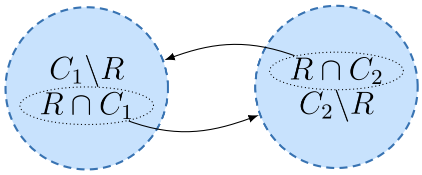


Fig. 1. *Representation of* $\mathcal{T}(C_1, C_2, R)$

- its right-hand side contains the sum of all the coefficients associated with a variable equal to 1 after the transformation.

A transformation $\mathcal{T} : \{C_1, C_2, R\} \mapsto \{C_1', C_2'\}$ is said to be *valid for* P, if there exist $K-2$ subsets $C_3, \ldots, C_K$ such that the characteristic vectors of the partitions $\pi_1 = \{C_1, C_2, C_3, \ldots, C_K\}$ and $\pi_2 = \{C_1', C_2', C_3, \ldots, C_K\}$ are in P.

3.2 Comparison of the two edge-representative formulations

To compare the linear relaxations of (F_{er}) and (F_{ext}), we consider $proj(R_{ext})$, the projection of (R_{ext}) onto the variable space of (R_{er}) and show that R_{ext} is strictly included in R_{er}.

First we prove that the linear relaxation obtained when considering formulation (F_{ext}) without Constraints (4) is not necessarily as good as the one from formulation (F_{er}).

Lemma 3.1 *$proj(R_{ext2}) \not\subset (R_{er})$ if $n \geq 4$ and $K \in \{2, \ldots, n-2\}$.*

Proof. To prove this, we identify a point of $proj(R_{ext2})$ which is not in (R_{er}).

Let $x^1 \in \mathbb{R}^{n(n-2)}$ be the incidence vector of a K-partition $\pi = \{C_1, \ldots, C_K\}$ such that $C_1 = \{1, 2, 3\}$ and $C_2 = \{4\}$. As represented in Table 2, we consider a second point x^2, identical to x^1 except for the values of the components related to nodes 1 to 4. We can check that x^2 is in (R_{ext2}) since it satisfies Constraints (1)-(3),(6) and (8)-(12). However, $r_3 + x_{2,3}$ is greater than 1. Since the projection of x^2 onto the space of the variables of (R_{er}) does not satisfy Constraints (4), it is not included in (R_{er}).

Vector	**Components**											
	r_2	r_3	r_4	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{2,3}$	$\tilde{x}_{2,3}$	$x_{2,4}$	$\tilde{x}_{2,4}$	$x_{3,4}$	$\tilde{x}_{3,4}$
x^2	0.2	0.4	0.4	0.8	0.6	0.6	0.8	0	0.8	0	0.6	0

Table 2
Component values related to nodes 1 to 4 for vectors x^2 of $\mathbb{R}^{n(n-2)}$.

□

When Constraints (4) are considered we can prove that, in all non-trivial cases, $proj(R_{ext})$ is strictly included in R_{er}.

Theorem 3.2 *$proj(R_{ext}) \subset R_{er}$ if $n \geq 4$ and $K \in \{2, \ldots, n-2\}$.*

Proof. Any point $x \in proj(R_{ext})$ satisfies Constraints (1)-(4) and (6). Constraints (8) and (10) show that Inequalities (5) are also satisfied. Hence $proj(R_{ext}) \subsetneq R_{er}$. To prove the strict inclusion, we exhibit a point in $R_{er} \setminus proj(R_{ext})$.

Let $x^1 \in \mathbb{R}^{\frac{n(n+1)}{2}-3}$ be the incidence vector of a K-partition $\{C_1, \ldots, C_K\}$ with $C_1 = \{1, 2, 3\}$. Let $x^2 \in \mathbb{R}^{\frac{n(n+1)}{2}-3}$ be a vector identical to x^1 except for components $x_{1,3}$ and $x_{2,3}$ which have value 0.5 (see Table 3).

Components	r_1	r_2	r_3	$x_{1,2}$	$x_{1,3}$	$x_{2,3}$
Initial point x^1	1	0	0	1	1	1
Modified point x^2	1	0	0	1	0.5	0.5

Table 3

Component values related to nodes 1, 2 *and* 3 *of vectors* x^1 *and* x^2.

Vector x^2 is in (R_{er}) since it satisfies Relations (1)-(6). To prove that it is also in $proj(R_{ext})$, we find a vector $\tilde{x}^2 \in \mathbb{R}^{\frac{n(n-1)}{2}}$ such that $((x^2)^T, (\tilde{x}^2)^T) \in (R_{ext})$. In particular, components $\tilde{x}_{1,3}$ and $\tilde{x}_{2,3}$ of $\tilde{x}^2$ must satisfy Constraints (8), (10) and (11) which respectively impose

$$r_3 + \tilde{x}_{1,3} + \tilde{x}_{2,3} = 1, \tag{24}$$

$$\tilde{x}_{1,3} \leq x_{1,3} \tag{25}$$

and

$$\tilde{x}_{2,3} \leq r_2. \tag{26}$$

Since r_2 and r_3 are equal to 0, $\tilde{x}_{2,3} = 0$ and $\tilde{x}_{1,3} = 1$ (according to (24) and (26)) but this implies that $x_{1,3} = 1$ which is different from 0.5.

□

This theorem ensures that the lower bound obtained with the linear relaxation (F_{ext}) is at least as good as the one obtained with (F_{er}). A numerical comparison of the quality of the linear relaxations of these formulations as well as formulations (F_{nc1}) and (F_{nc2}) is presented in Section 5.

3.3 Dimension of P_{ext}

In [3,4] we prove that P_{er} is full-dimensional when $K \in \{3, \ldots, n-2\}$. The following theorem shows that the same applies to P_{ext}.

Theorem 3.3 *The dimension of P_{ext} is equal to:*

(i) 0 *if* $K \in \{1, n\}$;
(ii) $n(n-2) + 2$ *if* $K = 2$;
(iii) $n(n-2)$ *if* $K \in \{3, 4, \ldots, n-2\}$ *(i.e., it is full dimensional);*
(iv) $\frac{n(n-1)}{2} - 1$ *if* $K = n - 1$.

Proof. If $K \in \{1, n\}$, then there is only one integer solution. If K is $n-1$, then there is only one integer solution for each edge $ij \in E$ (*i.e.*, $K-1$ clusters reduced to one node and one cluster is $\{i, j\}$). These solutions are affinely independent.

Suppose $K \in \{3, \ldots, n-2\}$. Assume that P_{ext} is included in a hyperplane $H = \{x \in \mathbb{R}^{n(n-2)} |\ \alpha^T x = \alpha_0\}$. We will prove that all its coefficients are equal to 0. Since K is in $\{3, \ldots, n-2\}$, a transformation $\mathcal{T}(C_1, C_2, R) \mapsto \{C_1', C_2'\}$ is valid for P_{ext} if C_1, C_2, C_1', C_2' are not empty (otherwise fewer than K clusters are obtained) and $|C_1 \cup C_2| \leq 4$ (otherwise no partition with K clusters can be obtained if K is equal to $n-2$).

Let a, b and c be three distinct nodes in V. Transformation $\mathcal{T}(\{a, b\}, \{c\}, \{b\})$ (represented in Figure 2a) gives

$$\tilde{\alpha}_{ab} + \alpha_{ab} = \tilde{\alpha}_{bc} + \alpha_{bc} \quad \forall a, b, c \in V. \tag{27}$$

We deduce from (27) that for all $a, b \in V$ the expression $\tilde{\alpha}_{ab} + \alpha_{ab}$ is equal to a scalar that we call β.

(a) $\mathcal{T}(\{a, b\}, \{c\}, \{b\})$. *(b)* $\mathcal{T}(\{1, b, c\}, \{d\}, \{c\})$.

Fig. 2. *Transformations used to prove the dimension of* (P_{ext}) *when* $K \in \{3, \ldots, n-2\}$.

For any three distinct nodes b, c and d in $V \backslash \{1\}$, the transformation represented in Figure 2b gives

$$\underbrace{\tilde{\alpha}_{1,c} + \alpha_{1,c}}_{=\beta} + \alpha_{bc} = \underbrace{\tilde{\alpha}_{c,d} + \alpha_{c,d}}_{=\beta}. \tag{28}$$

Thus, for any distinct nodes b and c greater than 1, α_{bc} is equal to 0. Since $\tilde{x}_{2,3}$ and $\tilde{x}_{1c}$ for all $c \in \{2, \ldots, n\}$ are artificial variables, the coefficients $\tilde{\alpha}_{1c}$ and $\tilde{\alpha}_{2,3}$ are null. Thus, $\tilde{\alpha}_{2,3} + \alpha_{2,3}$ is equal to 0 and the same applies to β and α_{1c}.

If K is equal to 2, Table 4 shows that P_{ext} is included in the $n-2$ independent hyperplanes induced by: $x_{1c} + x_{1,2} + 2\tilde{x}_{2c} - x_{2c} = 1$ for all $c \in \{3, \ldots, n\}$. In the following we prove that P_{ext} cannot be included in more than $n-2$ independent hyperplanes. To this end, we show that the coefficients of any hyperplane $H = \{x \in \mathbb{R}^{n(n-2)} |\ \alpha^T x = \alpha_0\}$ which includes P_{ext} are either equal to 0 or to a linear combination of the $n-2$ coefficients α_{2c} for all $c \in \{3, \ldots, n\}$.

When K is equal to 2 a transformation $\mathcal{T}(C_1, C_2, R) \mapsto \{C_1', C_2'\}$ is valid for P_{ext} if C_1, C_2, C_1' and C_2' are non-empty and if $C_1 \cup C_2 = V$. Let c be a node in $V \backslash \{1, 2\}$ and let V_1 and V_2 be two subsets such that $\{V_1, V_2, \{1, 2, c\}\}$ is a partition of V. From the transformations represented in Figures 3a and 3b we get

$$\alpha_{1c} + \alpha(c, V_1) = \tilde{\alpha}_{2c} + \alpha_{2c} + \alpha(c, V_2) \tag{29}$$

Configuration	Value of $x_{1c} + x_{1,2} + 2\tilde{x}_{2c} - x_{2c}$
$\{1, 2, c\} \subset C_1$	1
$\{1\} \subset C_1,\ \{2, c\} \subset C_2$	1
$\{1, 2\} \subset C_1,\ \{c\} \subset C_2$	1
$\{1, c\} \subset C_1,\ \{2\} \subset C_2$	1

Table 4
All possible configurations of nodes 1, 2 *and* $c \in V\backslash\{1, 2\}$ *when* $K = 2$.

and

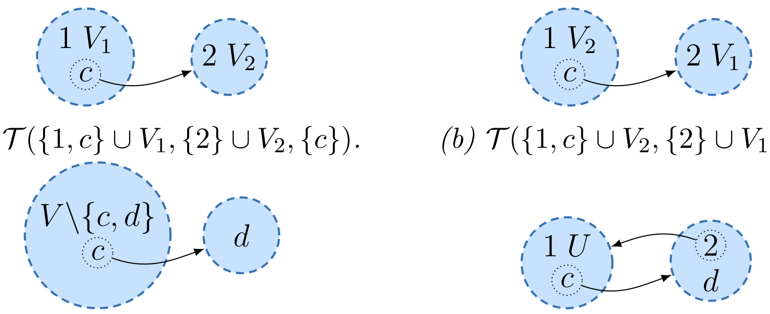
$$\alpha_{1c} + \alpha(c, V_2) = \tilde{\alpha}_{2c} + \alpha_{2c} + \alpha(c, V_1) \tag{30}$$

from which we deduce that

$$\alpha_{1c} = \tilde{\alpha}_{2c} + \alpha_{2c} \quad \forall c \in \{3, \ldots, n\} \tag{31}$$

and

$$\alpha(\{c\}, V_1) = \alpha(\{c\}, V_2) \quad \forall V_1, V_2 \text{ such that } V\backslash(V_1 \cup V_2) = \{1, 2, c\}. \tag{32}$$



(a) $\mathcal{T}(\{1, c\} \cup V_1, \{2\} \cup V_2, \{c\})$. *(b)* $\mathcal{T}(\{1, c\} \cup V_2, \{2\} \cup V_1, \{c\})$. *(c)* $\mathcal{T}(V\backslash\{d\}, \{d\}, \{c\})$. *(d)* $\mathcal{T}(\{1, c\} \cup U, \{2, d\}, \{2, c\})$.

Fig. 3. *Transformations used to prove the dimension of* (P_{ext}) *for* $K = 2$.

Let d be a node in V_1. By applying Equation (32) to $V_1' = V_1\backslash\{d\}$ and $V_2' = V_2 \cup \{d\}$ we deduce that α_{cd} is equal to 0 for all $c, d \in \{3, \ldots, n\}$.

Transformation $\mathcal{T}(V\backslash\{d\}, \{d\}, \{c\})$ (see Figure 3c) shows that

$$\alpha_{1c} + \alpha_{2c} + \underbrace{\alpha(c, V\backslash\{1, 2, c, d\})}_{=0} = \tilde{\alpha}_{cd} \quad \forall c, d \in \{3, \ldots, n\} \tag{33}$$

This result ensures that for all distinct nodes c and d in $V\backslash\{1, 2\}$, the expressions $\tilde{\alpha}_{cd}$ and $\alpha_{1c} + \alpha_{2c}$ are equal to a constant that we denote by γ. Thanks to Equation (31) for c equal to 3 and the fact that $\tilde{\alpha}_{2,3}$ is equal to 0, we deduce that the expression $2\alpha_{2,3}$ is also equal to γ.

Equations (31), (33) and the transformation represented in Figure 3d with $U = V\backslash\{1,2,c,d\}$ show that

$$\underbrace{\alpha_{1c}}_{=\gamma-\alpha_{2,c}} + \underbrace{\alpha(c,U)}_{=0} + \underbrace{\tilde{\alpha}_{2d} + \alpha_{2d}}_{=\alpha_{1d}} = \alpha_{1,2} + \alpha(2,U) + \underbrace{\tilde{\alpha}_{cd}}_{=\gamma} + \underbrace{\alpha_{cd}}_{=0} \tag{34}$$

which leads to

$$\alpha_{1d} = \alpha_{1,2} + \alpha(2,U) + \alpha_{2c}. \tag{35}$$

Finally, we use Equations (31), (33) and (35) to show that for all $c \in \{3,\dots,K\}$ the coefficients α_{1c}, $\tilde{\alpha}_{2c}$ and $\alpha_{1,2}$ are equal to the expressions $2\alpha_{2,3} - \alpha_{2c}$, $2(\alpha_{2,3} - \alpha_{2c})$ and $\alpha_{2,3} - \sum_{i\in\{4,\dots,n\}} \alpha_{2i}$, respectively. Thus, all the coefficients of H are either 0 or a linear combination of the $n-2$ coefficients α_{2c} for all $c \in \{3,\dots,n\}$. $\square$

3.4 The first linearization inequalities

Inequalities (10) involve artificial variables $\tilde{x}_{1j}$ when i is equal to 1 and artificial variable $\tilde{x}_{2,3}$ when j is equal to 3. We prove that in all other cases, these inequalities are facet-defining. Let $F^1_{i,j}$ be the face of (P_{ext}) induced by (10) for two nodes i, j such that $i < j$.

Theorem 3.4 *If $K \in \{3,\dots,n-2\}$, $i \geq 2$ and $j \geq 4$, $F^1_{i,j}$ is a facet of (P_{ext}).*

Proof. Let $H = \{x \in \mathbb{R}^{n(n-2)} \mid \alpha^T x = \alpha_0\}$ be a hyperplane including $F^1_{i,j}$. We prove that $\alpha^T x$ is necessarily equal to $\alpha_{ij}(x_{ij} - \tilde{x}_{ij})$.

To ensure that each transformation considered in the remainder of this proof is valid for $F^1_{i,j}$, we only consider configurations in which nodes i and j are not together in a cluster C of size greater than 2.

Let a, b and c be three nodes of V such that $|\{a,b,c\} \cap \{i,j\}| \in \{1,2\}$. Transformation $\mathcal{T}(\{a,b\},\{c\},\{b\})$ (represented in Figure 2a) shows that for any couple of nodes $\tilde{\alpha}_{ab} + \alpha_{ab}$ is equal to a constant that we denote by β.

Now let a, b and c be three nodes of $V\backslash\{1\}$ which satisfy $|\{a,b,c\}\cap\{i,j\}| \in \{1,2\}$ and $\{a,b\} \neq \{i,j\}$. Transformation $\mathcal{T}(\{1,a,b\},\{c\},\{b\})$ (represented in Figure 2b) proves that α_{ab} is equal to 0.

The fact that $\tilde{x}_{2,3}$ is an artificial variable ensures that $\alpha_{2,3} = \beta$. Since $\alpha_{2,3}$ is equal to 0, the same applies to β. Thus, all the coefficients are null except α_{ij} and $\tilde{\alpha}_{ij}$ which satisfy $\alpha_{ij} = -\tilde{\alpha}_{ij}$. $\square$

3.5 The second linearization inequalities

For any node $i \in \{4,\dots,n\}$ Inequality (11) can be reformulated by substituting artificial variables $\tilde{x}_{ij}$, r_i and $\tilde{x}_{1i}$ by their expression:

$$\tilde{x}_{ij} + x_{1,i} + \sum_{h=2}^{i-1} \tilde{x}_{hi} \leq 1. \tag{36}$$

Let $F^2_{i,j}$ be the face of (P_{ext}) induced by (36) for two nodes i, j such that $4 \leq i < j$.

Remark 3.5 $F^2_{i,j}$ is not facet-defining of P_{ext} if K is equal to $n-2$.

Proof. When K is equal to $n-2$ the clusters which are not reduced to one point are either two clusters of size 2 or one cluster of size 3.

Moreover, node i cannot be the only node in a cluster. Indeed, in that case the characteristic vector of the corresponding partition is not in the face induced by Equation (36) (since $\tilde{x}_{ij} + x_{1,i} + \sum_{h=2}^{i-1} \tilde{x}_{hi}$ is not equal to 1). Thus, $F^2_{i,j}$ is included in the hyperplane induced by

$$\sum_{k,l \in V \setminus \{i\}} x_{k,l} = 1. \tag{37}$$

□

Theorem 3.6 *If $K \in \{3, \ldots, n-3\}$ and $i \geq 4$, $F^2_{i,j}$ is a facet of (P_{ext}).*

Proof. Let $H = \{x \in \mathbb{R}^{n(n-2)} \mid \alpha^T x = \alpha_0\}$ be a hyperplane including $F^2_{i,j}$. We prove that $\alpha^T x$ is necessarily equal to $\alpha_{ij}(\tilde{x}_{ij} + x_{1i} + \sum\limits_{h=2}^{i-1} \tilde{x}_{hi})$.

To ensure that each transformation considered in the remainder of this proof is valid for $F^2_{i,j}$, we only consider configurations in which either nodes i and j are in the same cluster or i is not the representative of its cluster.

We consider a partition $\pi = \{C_1, \ldots, C_K\}$ such that $\{h, i\} \subset C_3$ with $h \in \{1, \ldots, i-1, j\}$. Similar to the proof of Theorem 3.3 – thanks to the transformations represented in Figure 2a and 2b – we prove that for any couple of distinct nodes a and b in $V \setminus \{i\}$ the coefficients α_{ab} and $\tilde{\alpha}_{ab}$ are equal to 0.

Let k and l be two nodes in $V \setminus \{1, i, j\}$. Transformation $\mathcal{T}(\{i,j\}, \{1,k,l\}, \{1,k\})$ represented in Figure 4a proves that

$$\underbrace{\alpha_{1l} + \alpha_{kl}}_{=0} + \tilde{\alpha}_{ij} = \alpha_{1i} + \alpha_{ik} + \underbrace{\alpha_{1j} + \alpha_{jk}}_{=0} \quad \forall k \in V \setminus \{1, i, j\}. \tag{38}$$

(a) $\mathcal{T}(\{i,j\}, \{1,k,l\}, \{1,k\})$. *(b)* $\mathcal{T}(\{i,j\}, \{k,l\}, \{l\})$.

Fig. 4. *Transformations used to prove the dimension of face $F^2_{i,j}$.*

For any node l less than i, the transformation represented in Figure 4b shows that

$$\tilde{\alpha}_{ij} + \underbrace{\tilde{\alpha}_{kl} + \alpha_{kl}}_{=0} = \tilde{\alpha}_{il} + \alpha_{il} + \underbrace{\tilde{\alpha}_{jl} + \alpha_{jl}}_{=0} \quad \forall l \in \{1, \ldots, i-1\}. \tag{39}$$

In particular, Equation (39) with l equals to 1 leads to $\tilde{\alpha}_{ij} = \alpha_{1i}$. This and Equation (38) give α_{ik} equals 0 for any node k in $V\backslash\{1,i,j\}$.

For l greater than i the transformation represented in Figure 4b shows that $\tilde{\alpha}_{il}$ is null. □

3.6 The third linearization inequalities

For any node $i \in \{4,\ldots,n\}$ Inequality (12) can be reformulated by substituting artificial variables r_i and $\tilde{x}_{1i}$ by their expression:

$$x_{ij} - \tilde{x}_{ij} - x_{1,i} - \sum_{h=2}^{i-1} \tilde{x}_{hi} \leq 0. \tag{40}$$

Let $F^3_{i,j}$ be the face of (P_{ext}) induced by (40) for two nodes i, j such that $4 \leq i < j$.

Remark 3.7 $F^3_{i,j}$ is not a facet of P_{ext} if K is equal to $n-2$.

Proof. We prove that the characteristic vector associated with any partition in $F^3_{i,j}$ satisfies

$$x_{hi} = \tilde{x}_{hi} \quad \forall h \in \{2,\ldots,i-1\}. \tag{41}$$

Since K is equal to $n-2$, the clusters which are not reduced to one point are either two clusters with two nodes or one cluster with three nodes.

If i and j are in the same cluster one can easily see that Equation (41) is satisfied. If i and j are in different clusters, node i is the representative of its cluster (otherwise, $x_{ij} - \tilde{x}_{ij} - x_{1,i} - \sum\limits_{h=2}^{i-1} \tilde{x}_{hi} = 0$ is not satisfied). Thus, x_{hi} and $\tilde{x}_{hi}$ are both equal to 0 for any node $h \in \{2,\ldots,i-1\}$.

□

Theorem 3.8 *If $K \in \{3,\ldots,n-3\}$ and $i \in \{4,\ldots,n\}$, $F^3_{i,j}$ is a facet of (P_{ext}).*

Proof. Let $H = \{x \in \mathbb{R}^{n(n-2)} \mid \alpha^T x = \alpha_0\}$ be a hyperplane including $F^3_{i,j}$. We prove that $\alpha^T x$ is necessarily equal to $\alpha_{ij}(x_{ij} - \tilde{x}_{ij} - x_{1i} - \sum\limits_{h=2}^{i-1} \tilde{x}_{hi})$.

To ensure that each transformation considered in the remainder of this proof is valid for $F^3_{i,j}$ we only consider configurations in which either nodes i and j are in the same cluster or node i is the representative of its cluster.

Let $\pi = \{C_1,\ldots,C_K\}$ be a partition such that $\{i,j\} \subset C_3$. Similar to the proof of Theorem 3.3 – thanks to the transformations represented in Figure 2a and 2b – we prove that for any couple of distinct nodes a and b in $V\backslash\{i,j\}$ the coefficients α_{ab} and $\tilde{\alpha}_{ab}$ are equal to 0.

Transformation $\mathcal{T}(\{i,j\},\{a\},\{j\})$ shows that $\tilde{\alpha}_{i,j} + \alpha_{i,j}$ is equal to $\tilde{\alpha}_{a,j} + \alpha_{a,j}$ for any node $a \in V\backslash\{i,j\}$. To prove that these two expressions are equal to 0, we consider a partition $\pi = \{\{1,j\},C_2,\{i\},C_4,\ldots,C_K\}$ and use transformation $\mathcal{T}(\{1,j\},C_2,\{1\})$.

The values of the remaining coefficients of H are obtained through the transformations presented in Table 5.

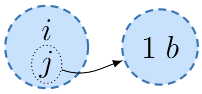
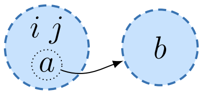
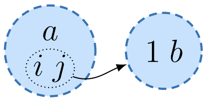
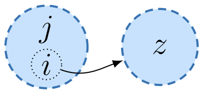
Conditions	Transformation	Result
$b \in V \backslash \{1, i, j\}$ $a \in \{1, \dots, i-1\} \backslash \{b\}$		$\underbrace{\tilde{\alpha}_{ij} + \alpha_{ij}}_{=0} = \underbrace{\alpha_{1j}}_{=0} + \alpha_{bj}$
		$\tilde{\alpha}_{ai} + \alpha_{ai} + \underbrace{\tilde{\alpha}_{aj} + \alpha_{aj}}_{=0} = \underbrace{\tilde{\alpha}_{ab} + \alpha_{ab}}_{=0} + \tilde{\alpha}_{ij}$
		$\underbrace{\tilde{\alpha}_{ai} + \alpha_{ai}}_{=\tilde{\alpha}_{ij}} + \underbrace{\tilde{\alpha}_{aj}}_{=\alpha_{1j}} + \underbrace{\alpha_{aj}}_{=0}$ $= \alpha_{bi} + \underbrace{\alpha_{1i}}_{=\tilde{\alpha}_{ij}} + \alpha_{1j} + \underbrace{\alpha_{bj}}_{=0}$
$\forall z \in \{i+1, \dots, n\} \backslash \{j\}$		$\underbrace{\tilde{\alpha}_{ij} + \alpha_{ij}}_{=0} = \tilde{\alpha}_{iz} + \underbrace{\alpha_{iz}}_{=0}$

Table 5

Transformations used in Theorem 3.8. Each line presents a step of the proof. The last column corresponds to the result.

□

3.7 The sub-representative inequalities

We introduce a new family of inequalities called the *sub-representative inequalities*. For each pair of nodes i and j with i less than j, the associated *sub-representative inequality* is defined as

$$x_{ij} \leq \sum_{h=1}^{i} \tilde{x}_{hj}. \tag{42}$$

This constraint ensures that if nodes i and j are in the same cluster, then j is represented by a node whose index is at most i. These inequalities are clearly valid for (P_{ext}). Let $F_{i,j}$ be the face of (P_{ext}) induced by (42) for two nodes i, j such that $i < j$.

Theorem 3.9 *If $K \in \{3, \dots, n-2\}$, $F_{i,j}$ is a facet of (P_{ext}) if and only if the following conditions are satisfied:*

(i) $i \geq 2$;

(ii) $K \leq n-3$ *or* $i = 2$.

Proof. We first prove that Equation (42) cannot be facet-defining if Conditions (i) and (ii) are not satisfied. If i is equal to 1, Constraint (42) is redundant since for all $k \in \{2, \dots, n\}$ $\tilde{x}_{1k}$ is an artificial variable equal to x_{1k}.

If Condition (ii) is not satisfied, K is equal to $n-2$ and i is greater than 2. In that case $F_{i,j}$ is included in the hyperplane induced by $x_{2j} = \tilde{x}_{2j}$. Indeed, if 2 and j are together in a cluster C, the same applies to i (otherwise the incidence vector of the K-partition is not in $F_{i,j}$). Since K is equal to $n-2$, node 1 cannot be in C and thus, node j is represented by node 2.

Let $H = \{x \in \mathbb{R}^{n(n-2)} \mid \alpha^T x = \alpha_0\}$ be a hyperplane including $F_{i,j}$ and let a and b be two distinct nodes in $V\backslash\{1,2,j\}$. We prove that $\alpha^T x$ is necessarily equal to $\alpha_{ij}(x_{ij} - \sum\limits_{h=1}^{i} \tilde{x}_{hj})$.

To ensure that each transformation considered in the remainder of this proof is valid for $F_{i,j}$ we only consider configurations in which node j is the only node of its cluster ($i.e$, $x_{ij} = \sum\limits_{h=1}^{i} \tilde{x}_{hj} = 0$) or nodes i and j are in the same cluster ($i.e$, $x_{ij} = \sum\limits_{h=1}^{i} \tilde{x}_{hj} = 1$). Let $\pi = \{C_1, \ldots, C_K\}$ be a K-partition such that:

- $\{1, 2, a, b\} \subseteq C_1 \cup C_2$;
- $C_3 = \{j\}$;
- the remaining nodes are scattered in clusters C_2 and C_4 to C_K.

Let U be the set $C_2\backslash\{1,2,a,b\}$. Transformations $\mathcal{T}(\{1,a,b\}, \{2\} \cup U, \{b\})$ and $\mathcal{T}(\{1,b\}, \{2,a\} \cup U, \{b\})$ show that

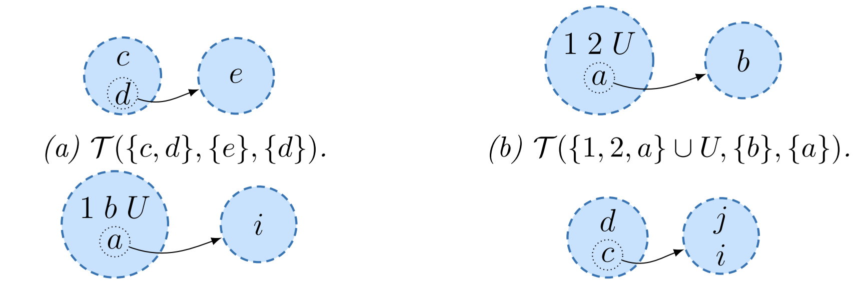
$$\alpha_{ab} = 0 \quad \forall a, b \in V\backslash\{1,2,j\}. \tag{43}$$

Let c, d and e be three distinct nodes in $V\backslash\{i,j\}$. The transformation represented in Figure 5a gives

$$\tilde{\alpha}_{cd} + \alpha_{cd} = \beta \quad \forall c, d \in V\backslash\{i,j\}, \tag{44}$$

with β a scalar. In particular, we obtain:

- $\alpha_{1k} = \beta$ for all $k \in V\backslash\{i,j\}$ (since $\tilde{x}_{1k}$ is an artificial variable);
- $\tilde{\alpha}_{cd} = \beta$ for all $c, d \in V\backslash\{1,2,i,j\}$ thanks to Equation (43).



(a) $\mathcal{T}(\{c,d\}, \{e\}, \{d\})$. *(b)* $\mathcal{T}(\{1,2,a\} \cup U, \{b\}, \{a\})$.

(c) $\mathcal{T}(\{1,a,b\} \cup U, \{i\}, \{a\})$. *(d)* $\mathcal{T}(\{c,d\}, \{i,j\}, \{c\})$.

Fig. 5. *Transformations used to prove that* $F_{i,j}$ *is a facet if Conditions* (i), (ii) *and* (iii) *are satisfied.*

For all a in $V\backslash\{1,2,i,j\}$ transformation $\mathcal{T}(\{1,2,a\} \cup U, \{b\}, \{a\})$ (see Figure 5b) proves that α_{2a} is null and the transformation represented in Figure 5c shows that $\tilde{\alpha}_{ai} + \alpha_{ai} = \beta$. Transformations $\mathcal{T}(\{1\} \cup U, \{2, i\}, \{i\})$ and $\mathcal{T}(\{1\} \cup U, \{2, i\}, \{2\})$ prove that $\tilde{\alpha}_{2i} + \alpha_{2i}$ and α_{1i} are equal to β, respectively. Since β is equal to $\tilde{\alpha}_{2,3} + \alpha_{2,3}$ and $\tilde{x}_{2,3}$ is an artificial variable, we deduce that β is equal to 0.

Transformation $\mathcal{T}(\{c, i\}, \{j\}, \{i\})$ leads to $\alpha_{ij} = -\tilde{\alpha}_{ij}$. Transformation $\mathcal{T}(\{c, d\}, \{i, j\}, \{c\})$ (see Figure 5d) gives $\alpha_{cj} = 0$ if c is greater than i and $\tilde{\alpha}_{ij} = \tilde{\alpha}_{cj} + \alpha_{cj}$ otherwise. We then prove thanks to $\mathcal{T}(\{c, d, i\}, \{j\}, \{c\})$ that for all $c > i$ $\tilde{\alpha}_{cj} = 0$.

Lastly, for $i > 2$ Conditions (iii) ensure that K is less than $n - 2$. Thus, transformation $\mathcal{T}(\{1, c, d, i\}, \{j\}, \{d, j\})$ leads to $\alpha_{cj} = 0$ for all $c \in \{2, \ldots, i - 1\}$.

$\square$

3.8 At least K clusters or at most K clusters

Formulation (F_{ext}) can easily be adapted to the cases in which at least or at most K clusters are sought, by replacing Equality (6) with

$$\sum_{i=1}^{n} r_i \leq K \tag{45}$$

or

$$\sum_{i=1}^{n} r_i \geq K. \tag{46}$$

This observation leads to the two following formulations:

$$(F^+_{ext}) \begin{cases} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to } (1) - (4), (8) - (12) \text{ and } (46) \end{cases}$$

and

$$(F^-_{ext}) \begin{cases} \text{minimize} & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to } (1) - (4), (8) - (12) \text{ and } (45) \end{cases}$$

The variable substitutions considered in (F_{ext}) are also possible for these two formulations with the exception of r_3, as we no longer have (6).

Let $P_{ext,K}$, $P^-_{ext,K}$ and $P^+_{ext,K}$ denote the convex hulls of all feasible solutions of (F_{ext}), (F^+_{ext}) and (F^-_{ext}), respectively. Let P_1 be $P_{ext,K}$ or one of its faces and let P_2 be $P^+_{ext,K}$, $P^-_{ext,K}$ or one of their faces.

Remark 3.10 If $P_1 \subset P_2$ and P_2 contains a feasible solution π with a number of clusters different from K, then $dim(P_2) \geq dim(P_1) + 1$.

This is due to the fact that $P_{ext,K}$ is a projection of either $P^+_{ext,K}$ or $P^-_{ext,K}$ on the hyperplane defined by $\sum_{i=1}^{n} r_i = K$.

The following proofs are based on this remark and the results obtained previously with P_{ext}.

Theorem 3.11 *The dimension of P^+_{ext} is equal to :*

(i) $n(n-2)+1$ *if* $K \in \{1, \dots, n-2\}$;

(ii) $\frac{n(n-1)}{2}$ *if* $K = n-1$.

Proof. Polytope $P^+_{ext,n-1}$ only contains the $\frac{n(n-1)}{2}$ affinely independent solutions of $P_{ext,n-1}$ and the solution $\pi^n = \{\{1\}, \{2\}, \dots, \{n\}\}$. For k less than $n-1$, $P^+_{ext,k}$ includes π^n and all the solutions in $P_{ext,n-2}$. □

Theorem 3.12 *The dimension of P^-_{ext} is equal to :*

(i) 0 *if* $K = 1$;

(ii) $n(n-2)+3$ *if* $K = 2$;

(iii) $n(n-2)+1$ *if* $K \in \{3, \dots, n\}$.

Proof. Polytope $P^-_{ext,1}$ only contains one solution.

$P^-_{ext,K}$ for $K \in \{3, \dots, n\}$ contains all the solutions in $P_{ext,n-2}$ and $\pi^1 = \{\{1, 2, \dots, n\}\}$.

$P^-_{ext,2}$ contains $P_{ext,2}$ and solution π^1 so $dim(P^-_{ext,2}) \geq n(n-2)+3$. Moreover, as represented in Table 4, $P^-_{ext,2}$ is included in the hyperplanes $x_{1c} + x_{1,2} + 2\tilde{x}_{2c} - x_{2c} = 1$ for all $c \in \{3, \dots, n\}$ which ensure that $dim(P^-_{ext,2}) = n(n-2)+3$. □

We now prove that all the facets identified previously for $P_{ext,K}$ are also facets of $P^+_{ext,K}$ and $P^-_{ext,K}$.

Let F^{1+}_{ij} and F^{1-}_{ij} respectively be the faces of $P^+_{ext,K}$ and $P^-_{ext,K}$ associated with Inequality (10) for $i, j \in V$ and $K \in \{1, \dots, n\}$.

Theorem 3.13 *If $K \in \{3, \dots, n\}$, $i \geq 2$ and $j \geq 4$, the face $F^{1+}_{i,j}$ is a facet of $P^+_{ext,K}$.*

Proof. All solutions in F^1_{ij} are also in $F^{1+}_{i,j}$. Moreover, $F^{1+}_{i,j}$ contains any solution with $K+1$ clusters such that i and j are not in the same clusters. □

The following theorems can be proved using similar arguments.

Theorem 3.14 *If $K \in \{1, \dots, n-2\}$, $i \geq 2$ and $j \geq 4$, the face $F^{1-}_{i,j}$ is a facet of $P^-_{ext,K}$.*

Let $F^{2+}_{i,j}$ and $F^{2-}_{i,j}$ respectively be the faces of $P^+_{ext,K}$ and $P^-_{ext,K}$ induced by (36) for two nodes i, j such that $4 \leq i < j$.

Theorem 3.15

- *If $K \in \{3, \dots, n\}$ and $i \geq 4$, $F^{2+}_{i,j}$ is a facet of $P^+_{ext,K}$.*
- *If $K \in \{1, \dots, n-3\}$ and $i \geq 4$, $F^{2-}_{i,j}$ is a facet of $P^-_{ext,K}$.*

Let $F^{3+}_{i,j}$ and $F^{3-}_{i,j}$ respectively be the faces of $P^+_{ext,K}$ and $P^-_{ext,K}$ induced by (40) for two nodes i, j such that $4 \leq i < j$.

Theorem 3.16

- *If* $k \in \{3, \dots, n\}$ *and* $i \in \{4, \dots, n\}$, $F_{i,j}^{3+}$ *is a facet of* $P_{ext,K}^{+}$.
- *If* $k \in \{1, \dots, n-3\}$ *and* $i \in \{4, \dots, n\}$, $F_{i,j}^{3-}$ *is a facet of* $P_{ext,K}^{-}$.

Let $F_{i,j}^{+}$ and $F_{i,j}^{-}$ respectively be the faces of $P_{ext,K}^{+}$ and $P_{ext,K}^{-}$ induced by (42) for two nodes i, j such that $i < j$.

Theorem 3.17

- *If* $K \in \{3, \dots, n\}$, $F_{i,j}^{+}$ *is a facet of* $P_{ext,K}^{+}$ *if the following conditions are satisfied:*
  - *(i)* $i \geq 2$;
  - *(ii)* $K \leq n-3$ *or* $i = 2$.
- *If* $K \in \{1, \dots, n-2\}$, $F_{i,j}^{-}$ *is a facet of* $P_{ext,K}^{-}$ *if the following conditions are satisfied:*
  - *(i)* $i \geq 2$;
  - *(ii)* $K \leq n-3$ *or* $i = 2$.

4 Branch-and-cut strategy

In this section, we briefly present our branch-and-cut algorithm which is divided in two steps:

(i) a thorough *cutting-plane step* at the root node during which valid inequalities are added to an incomplete version of formulation (F_{ext}) (i.e., formulation (F_{ext}) without its largest families of inequalities) in order to improve as much as possible the value of the relaxation at the root of the branch-and-cut tree;

(ii) a classical CPLEX *branch-and-cut step* which starts with the full formulation (F_{ext}) plus the inequalities generated during the cutting-plane step which were tight for the last computed relaxation.

Our branch-and-cut takes advantage of the following families of inequalities that we considered in a previous work [3,4]: the *strengthened triangle inequalities* [3,4], the 2-*partition inequalities* [22], the *general clique inequalities* [12] and the *paw inequalities* [4].

In [3,4] we proved that the triangle inequalities (2) are not facet-defining for P_{er} when i is greater than both j and k [3,4]. In that case they are dominated by the following inequalities which are all facet-defining for $K \in \{2, \dots, n-3\}$:

$$x_{ij} + x_{ik} - x_{jk} + r_i \leq 1 \quad \forall j, k \in V \backslash \{i\},\ j < k < i. \tag{47}$$

We directly use this reinforcement in formulations (F_{er}) and (F_{ext}).

The 2-*partition inequality* associated with two disjoint subsets of nodes S and T is:

$$x(S,T) - x(S) - x(T) \leq \min(|S|, |T|). \tag{48}$$

Given a subset $Z \subset V$ of size $qK + p$ (with $p \in \{0, 1, \dots, K-1\}$) the *general clique inequality* associated with Z is:

$$x(Z) \geq \frac{(q+1)q}{2}p + \frac{q(q-1)}{2}(K-p). \tag{49}$$

The *paw inequality* related to four nodes a, b, c and d of V is:

$$x_{ab} + x_{bc} - x_{ac} + x_{cd} + x_b + x_c \leq 2. \tag{50}$$

We use two approaches to separate these inequalities:

(i) greedy algorithms which quickly search for several violated inequalities;

(ii) Kernighan-Lin type algorithms [29] which are slower, only seek to find one inequality and are only used when all the greedy algorithms fail at finding any violated inequality.

We use the last approach only for the 2-*partition inequalities* and the *general clique inequalities*, which appeared to be significantly more useful in our previous studies [2,3,4].

Our greedy algorithms to separate the 2-partition inequalities is based on Grötschel and Wakabayashi [22]. It seeks inequalities in which set S is reduced to only one node.

For the *general clique inequalities*, we adapt a greedy algorithm which achieves an approximation factor of 2 of the densest at least k-subgraph problem [30]. This problem consists in identifying a subgraph of at least k nodes of maximal density (*i.e.,* a set $S \subset V$ which maximizes $\sum\limits_{i,j \in S} w_{i,j}$). The worst case running time of this algorithm is $\mathcal{O}(n^3)$ but, as mentioned in [22], it is much faster in practice.

We define a greedy heuristic for the *paw inequalities* which for each b successively seeks the nodes c, a and d which maximize the left-hand side of Equation (50).

For the *sub-representative inequalities* we consider each node $i \in V$ and seek the node j which leads to the most violated inequality. The running time of these two greedy algorithms is $\mathcal{O}(n^2)$. The complexity of each phase of a Kernighan-Lin algorithm is also $\mathcal{O}(n^2)$. In practice, it appears that the number of phases is rather small (around 3 in our experiments).

In order to quickly compute the successive relaxations during the cutting-plane step, we remove Constraints (2), (4), (10)-(12) and (47) from (F_{ext}). At each cutting-plane step, violated inequalities from these removed families are sought through enumeration heuristics. For example, to separate the triangle inequalities, three permutations of $(1, 2, \dots, n)$ – $S_i = (s_1^i, s_2^i, \dots, s_n^i)$, $S_j = (s_1^j, s_2^j, \dots, s_n^j)$ and $S_k = (s_1^k, s_2^k, \dots, s_n^k)$ – are randomly generated. These three sets define the order in which the triangle inequalities are tested (e.g., the first inequality considered corresponds to nodes s_1^i, s_1^j and s_1^k). The enumeration heuristics used for the other families of inequalities follow the same principle. To limit the number of inequalities in the formulation, we only add for each family the 500 most violated inequalities among the 3000 first that we find.

During the cutting-plane step, feasible solutions are regularly obtained from linear relaxations via a greedy algorithm which first identifies the K highest representative variables of the current linear relaxation x^* and then assigns each non-representative node i to the cluster of the representative r which maximizes x^*_{ir}.

When no violated inequality is found, the branch-and-cut step is initiated with the complete formulation (F_{ext}) plus the inequalities generated during the cutting-plane step which are tight for the last relaxation. This step uses CPLEX branch-and-cut in which all the greedy separation algorithms are used at each node of the enumeration tree. The Kernighan-Lin type algorithms are not considered during this step due to their complexity.

5 Numerical results

In this section we compare the four formulations presented in Section 2 in terms of the value of the linear relaxation and computation time. The performance of the branch-and-cut algorithm described in Section 4 is matched against the default CPLEX branch-and-cut approach for each of the four formulations.

5.1 Quality of the linear relaxations

The linear relaxations of the formulations are compared over data sets D_1, D_2 and D_3 each composed of 100 different graphs for each considered value of n. In these data sets the values of the edge weights are randomly generated using a uniform distribution respectively in the following intervals: $[0, 500]$, $[-250, 250]$ and $[-500, 0]$. These instances are intended to be rather difficult to solve when compared to real instances.

For a given graph G and a given formulation F, let x^* be the value of the optimal integer solution and let x^r be the value of the corresponding linear relaxation. We define the *relative gap* of F over G by $100 * |x^* - x^r|/x^*$. The smaller the relative gap, the better the linear relaxation.

The results obtained for the four formulations over D_1, D_2 and D_3 are presented in Table 6, Table 7 and Table 8, respectively. They contain, for each couple (n, K) and each formulation, the average relative gap of the 100 corresponding graphs.

We first observe that the values of the linear relaxations are significantly different in the three tables. The highest values are obtained with the instances from D_1 whereas the lowest are observed over the instances of D_3. This is likely related to the fact that the K-partitioning problem is polynomial whenever all the edges have negative values [21]. The main conclusion of these three tables is that the extended formulation always gives the best results closely followed by (F_{er}). The node-cluster formulations are significantly worse except for (F_{nc2}) in D_3 and (F_{nc1}) in D_1 when K is equal to 2. In an effort to explain these observations, we prove (see Appendix) the three following bounds.

Bound 5.1 *The optimal value of the linear relaxation of* (F_{nc2}) *is less than or equal to* $\min_{j\in\{2,\ldots,K\}} \min_{i<j} \frac{w_{ij}}{2^{K-j+1}}$.

Bound 5.2 *If* $K \in \{3,\ldots,n\}$*, then the optimal value of the linear relaxation of* (F_{nc1}) *is less than or equal to* $\min_{j\in\{2,\ldots,K-1\}} \min_{i<j} \frac{w_{ij}}{2^{K-j}}$.

Bound 5.3 *If* K *is equal to* 2*, then the optimal value of the linear relaxation of* (F_{nc1}) *is in the interval* $[\min_{i,j\in V} w_{ij}\frac{n-1}{2}, \frac{1}{2}\sum_{i=2}^{n} w_{1i}]$.

These bounds show that the values of the linear relaxation, for these two formulations, are extremely weak over instances with positive weights on their edges (*e.g.*, instances from D_1). To illustrate this statement, we consider an instance G_e for which all edge weights are equal to 1. Bound 5.1 states that the linear relaxation of (F_{nc2}) could not be better than $\frac{1}{2^{K-1}}$. This shows why the quality of the relaxation decreases with K. We can draw similar conclusions for (F_{nc1}) via Bound 5.2 which gives a maximal value of the relaxation equal to $\frac{1}{2^{K-2}}$ when K is greater than 2. Finally, the last bound explains why the results of (F_{nc1}) are slightly better when K is equal to 2. Indeed, in this case, Bound 5.3 shows that the value of the linear relaxation of G_e is equal to $\frac{n-1}{2}$.

5.2 Optimal resolution

In order to compare the four formulations in terms of optimal resolution, we use for each of them CPLEX 12.7 with the default parameters as well as a 2.50 GHz Intel Core i5 2520M CPU equipped with 8 GByte RAM. The n general clique inequalities induced by the sets Z of size $n-1$ and n are used to reinforce each formulation since it proved to significantly speed up the resolution in our preliminary experiment. The branch-and-cut algorithm described in Section 4 is also considered.

5.2.1 Random instances

For a given value of n and K, the five resolution methods (*i.e.*, the four formulations and the branch-and-cut algorithm) are tested on the 10 first instances of data sets D_1, D_2 and D_3. We interrupt the cutting-plane step after 500 seconds as we observed that the increase of the relaxation value tends to decrease over the iterations.

A total of 2400 partitioning problems are solved on the random instances. Therefore, the maximal computation time considered is relatively small (10 minutes, for a total computing time of 400 hours).

The results over the three data sets are presented in Tables 10, 11 and 12. We first observe that the performances of formulations (F_{er}) and (F_{ext}) are very similar in terms of time and gap. The better quality of the linear relaxation of (F_{ext}) is compensated by a faster computation of the linear relaxation of (F_{er}) which enables us to explore more nodes in the branch-and-cut tree. This

n	Formulation	K								
		2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
15	(F_{nc1})	82	97	98	99	99	99	99	100	100
	(F_{nc2})	99	99	99	99	100	100	100	100	100
	(F_{er})	87	79	71	60	48	34	20	11	5
	(F_{ext})	76	70	61	51	39	25	13	7	3
16	(F_{nc1})	83	98	99	99	99	99	99	100	100
	(F_{nc2})	99	99	99	100	100	100	100	100	100
	(F_{er})	88	82	74	64	53	41	24	14	8
	(F_{ext})	78	73	65	55	44	32	18	9	5
17	(F_{nc1})	84	98	99	99	99	100	100	100	100
	(F_{nc2})	99	99	99	100	100	100	100	100	100
	(F_{er})	89	83	76	68	58	47	31	18	11
	(F_{ext})	79	74	67	59	49	38	23	12	7
18	(F_{nc1})	86	98	99	99	99	100	100	100	100
	(F_{nc2})	99	99	100	100	100	100	100	100	100
	(F_{er})	90	85	79	71	62	51	39	26	16
	(F_{ext})	81	77	70	62	53	42	30	18	10
19	(F_{nc1})	86	98	99	99	99	100	100	100	100
	(F_{nc2})	99	99	100	100	100	100	100	100	100
	(F_{er})	91	86	81	74	67	58	47	34	21
	(F_{ext})	82	78	72	65	57	49	38	25	14
20	(F_{nc1})	87	99	99	99	100	100	100	100	100
	(F_{nc2})	99	100	100	100	100	100	100	100	100
	(F_{er})	92	88	83	77	70	62	52	39	25
	(F_{ext})	83	80	75	69	61	53	44	31	18

Table 6

Average relative gap percentage of the four considered formulations over one hundred graphs of D_1 instances for different values of n and K.

suggests that a more dynamic management of the constraints in (F_{ext}) may improve the results.

For each of the three data sets, the efficiency of the node-cluster formulations generally decreases when K increases. In particular, when K is equal to 2, (F_{nc1}) is significantly better. The main exception to these observations appears in data set D_2 when n is equal to 50 where these formulations fail at finding good feasible solutions.

Data sets D_1 and D_2 lead to similar observations. Method (BC) provides reduced computation times and better gaps than (F_{er}) and (F_{ext}) while the

n	Formulation	K								
		2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
15	(F_{nc1})	122	83	79	83	92	105	123	147	184
	(F_{nc2})	118	73	61	56	53	51	50	49	49
	(F_{er})	29	9	8	9	12	15	17	20	24
	(F_{ext})	21	8	7	9	11	12	14	13	14
16	(F_{nc1})	128	88	85	87	95	107	123	144	175
	(F_{nc2})	124	78	66	60	57	56	55	54	54
	(F_{er})	30	9	8	9	11	14	17	20	24
	(F_{ext})	22	8	8	9	11	13	14	15	16
17	(F_{nc1})	149	97	91	94	101	111	125	143	167
	(F_{nc2})	146	86	73	67	64	62	61	59	58
	(F_{er})	41	13	10	12	14	16	19	22	25
	(F_{ext})	32	11	10	11	13	15	16	17	18
18	(F_{nc1})	146	100	95	97	103	112	124	139	159
	(F_{nc2})	143	90	77	70	67	64	63	61	60
	(F_{er})	38	14	11	12	14	17	19	21	23
	(F_{ext})	29	12	11	12	14	16	17	18	18
19	(F_{nc1})	147	107	102	104	110	118	129	143	162
	(F_{nc2})	143	96	84	77	74	72	70	69	68
	(F_{er})	37	16	14	15	17	20	22	25	28
	(F_{ext})	29	15	14	14	16	18	20	21	21
20	(F_{nc1})	148	111	107	109	115	123	134	146	163
	(F_{nc2})	144	100	88	83	79	77	76	74	73
	(F_{er})	37	18	16	17	19	21	23	25	28
	(F_{ext})	29	17	16	17	18	20	22	23	23

Table 7

Average relative gap percentage of the four considered formulations over one hundred graphs of D_2 instances for different values of n and K.

node-cluster formulations give the worst ones. The size of the branch-and-cut tree for (F_{nc1}) and (F_{nc2}) shows that a lot of branching is necessary to improve the quality of the linear relaxation. This can be due to a poor initial value of the relaxation as well as to the presence of symmetry in the formulations.

In data set D_3 formulation (F_{nc1}) gives the best results for K equal to 2 and the worst ones otherwise, while (F_{nc2}) is better than (F_{ext}) and (F_{er}). This is also the data set in which the node-cluster formulations browse the fewest nodes while the edge-representative formulations browse the most nodes. The results of our branch-and-cut approach on this data set are almost always

n	Formulation	K								
		2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
15	(F_{nc1})	10	22	36	54	76	105	143	196	274
	(F_{nc2})	3	7	13	18	26	34	44	56	72
	(F_{er})	3	8	13	19	26	34	44	57	73
	(F_{ext})	3	7	13	18	25	34	44	56	72
16	(F_{nc1})	9	20	33	49	69	94	127	171	232
	(F_{nc2})	3	7	12	17	23	31	40	51	65
	(F_{er})	3	7	12	17	23	31	40	51	66
	(F_{ext})	3	7	11	17	23	31	40	51	64
17	(F_{nc1})	9	19	32	46	64	86	115	151	201
	(F_{nc2})	3	7	11	16	22	29	37	47	59
	(F_{er})	3	7	11	16	22	29	37	47	59
	(F_{ext})	3	7	11	16	22	29	37	46	58
18	(F_{nc1})	8	18	30	43	60	80	104	135	176
	(F_{nc2})	3	6	10	15	21	27	35	43	53
	(F_{er})	3	6	11	15	21	28	35	43	54
	(F_{ext})	3	6	10	15	21	27	34	43	53
19	(F_{nc1})	8	17	28	40	56	74	96	123	158
	(F_{nc2})	3	6	10	14	20	25	32	40	49
	(F_{er})	3	6	10	14	20	26	32	40	50
	(F_{ext})	3	6	10	14	19	25	32	40	49
20	(F_{nc1})	8	16	27	39	53	70	90	114	145
	(F_{nc2})	3	6	10	14	19	25	31	38	46
	(F_{er})	3	6	10	14	19	25	31	38	47
	(F_{ext})	3	6	10	14	19	25	31	38	46

Table 8

Average relative gap percentage of the four considered formulations over one hundred graphs of D_3 instances for different values of n and K.

worse than the ones of the other methods. This is not surprising as it has been noticed experimentally [3] that a remarkably small number of violated inequalities can be found for D_3 instances.

In the light of these results it appears that, for these difficult instances, the node-cluster formulations give better results for D_3 instances (the polynomial case), whereas the edge-representative formulations should be preferred otherwise.

Instance	**1**	**2**	**3**	**4**	**5**	**6**
Average weight	-32	-354	162	-29	-349	163
Standard deviation	9443	10331	10194	9428	10332	10207
Minimal weight	-151693	-212231	-181596	-153151	-214271	-183342
Maximal weight	239752	170713	220276	242057	172355	222394

Table 9

Average, standard deviation, minimal and maximal value of the edge weights for the six instances from the spin-glass data set.

5.2.2 *Instances from the literature*

We now evaluate the behavior of the five methods on instances from the literature with an increased computation time of 1 hour. Eight instances from the TSPLIB [38] with different number of cities (from 29 to 127) as well as six instances of ground-states of the Ising model of spin-glass with 100 nodes [33,34] are considered. The results of the TSPLIB and the spin-glass instances are presented in Table 13 and Table 14, respectively.

Similar to D_1 instances, the weights in the TSPLIB are non-negative as they correspond to distances between cities. This may explain why the results look similar. The node-cluster formulations give for most instances the worst results while the performances of (F_{er}) and (F_{ext}) are similar. The branch-and-cut algorithm results are significantly better and it always finds the optimal solution for instances with less than 70 nodes. We can observe that all methods show better performances on this data set than over the random instances of D_1. A possible explanation is the fact that real instances are more likely to contain underlying clusters (*e.g.,* several cities close to one another in a TSPLIB instance) than random ones.

Like dataset D_2, the spin-glass instances contain positive and negative values on their edges (see Table 9). The four formulations behave similarly in both data sets, however, the results of the branch-and-cut are not as impressive. Indeed, for K equal to 4 and 6 it only provides a slight improvement compared to our formulations and for K equal to 8 the results are worst. We believe that this is due to the huge number of inequalities (in particular the $\mathcal{O}(n^3)$ triangle inequalities) which slows down the separation algorithms. Thus, interesting perspectives to improve the performances on large graphs would be to enhance the separation algorithms as well as improving the management of the triangle inequalities.

6 Conclusion

We have compared two node-cluster formulations for the K-partitioning problem with an edge-representative formulation (F_{er}) and its extended version (F_{ext}). An advantage of the formulations with representative variables is that they enable us to fix the number K of clusters without resorting to the *node-cluster variables*, and thus, avoiding a lot of symmetry. This advantage is reflected in our numerical experiments when weights are not all negative and K is strictly greater than 2.

These numerical results can be improved significantly by using facet-defining inequalities in a branch-and-cut process that takes advantage of facet-defining inequalities previously studied in [3,4] as well as the *sub-representative inequalities* introduced in this paper.

Finally, our extended formulation (F_{ext}) has an even better relaxation than our basic edge-representative formulation but at the price of an increased size. As a result, both formulations give similar performances in our experiments. We believe that the better relaxation provided by (F_{ext}) combined with an efficient management of the triangle inequalities might significantly improve the results for larger instances.

n	K	Time (s) and Gap (%)										Nodes				
		(F_{nc1})		(F_{nc2})		(F_{er})		(F_{ext})		(BC)		(F_{nc1})	(F_{nc2})	(F_{er})	(F_{ext})	(BC)
		time	gap	time	gap	time	gap	time	gap	time	gap					
20	**2**	2	**0**	8	**0**	6	**0**	7	**0**	**0**	**0**	948	1656	12	5	0
	4	161	1	550	6	16	**0**	24	**0**	**2**	**0**	71059	26046	460	293	0
	6	328	3	36	**0**	6	**0**	9	**0**	**0**	**0**	84519	6094	344	144	0
	8	394	7	2	**0**	1	**0**	1	**0**	**0**	**0**	167181	474	45	16	0
30	**2**	170	**0**	600	38	156	**0**	135	**0**	**3**	**0**	43224	4273	247	102	0
	4	600	61	600	72	601	9	600	9	**149**	**0**	45047	2445	2728	1723	1
	6	601	84	601	61	481	6	496	6	**59**	**0**	52824	3846	4547	2500	46
	8	601	93	601	40	88	**0**	121	**0**	**8**	**0**	73342	7740	2139	1346	2
40	**2**	600	25	600	59	600	20	600	11	**35**	**0**	13432	1645	199	104	0
	4	601	84	600	83	606	38	607	38	**615**	**8**	15754	676	29	3	0
	6	601	95	600	78	609	38	604	34	**614**	**13**	10452	711	565	226	0
	8	601	97	600	71	601	34	601	34	**556**	**15**	9749	910	1598	624	0
50	**2**	600	39	601	79	600	11	600	11	**600**	**5**	3402	286	16	10	0
	4	601	90	601	88	611	40	604	51	**604**	**20**	4090	75	0	0	0
	6	601	97	604	84	628	49	611	52	**600**	**28**	2585	79	0	0	0
	8	601	99	601	79	609	45	604	56	**613**	**29**	5679	171	0	0	0

Table 10

Average results (in terms of time, gap and number of nodes in the branch-and-cut tree) obtained for each of the five methods over 10 *instances of D_1. (BC) corresponds to the branch-and-cut algorithm presented in Section 4. Bold results are the best in terms of gap and time. A gap of* 0% *corresponds to an optimal solution over each of the 10 instances.*

n	K	Time (s) and Gap (%)										Nodes				
		(F_{nc1})		(F_{nc2})		(F_{er})		(F_{ext})		(BC)		(F_{nc1})	(F_{nc2})	(F_{er})	(F_{ext})	(BC)
		time	gap	time	gap	time	gap	time	gap	time	gap					
20	**2**	1	**0**	13	**0**	8	**0**	8	**0**	**0**	**0**	32	272	59	8	0
	4	31	**0**	87	**0**	10	**0**	15	**0**	**5**	**0**	10912	5948	25	27	1
	6	244	1	97	**0**	14	**0**	16	**0**	**11**	**0**	42353	6841	67	60	30
	8	526	19	63	**0**	**10**	**0**	**10**	**0**	13	**0**	64315	6273	211	142	97
30	**2**	14	**0**	308	6	259	3	145	**0**	**4**	**0**	2608	2743	1040	94	0
	4	600	109	600	130	366	12	443	19	**112**	**0**	16549	1347	1352	905	0
	6	600	130	600	123	400	11	407	18	**184**	**0**	9372	2269	885	668	2
	8	601	158	600	121	409	15	419	15	**275**	**1**	6197	3174	1097	869	2
40	**2**	600	80	601	197	600	69	600	70	**253**	**0**	12495	380	245	107	0
	4	600	465	600	488	618	195	608	195	**623**	**39**	3778	128	8	2	0
	6	600	603	600	569	607	244	608	245	**651**	**45**	1076	247	3	1	0
	8	600	1286	601	1134	601	555	603	556	**644**	**60**	522	404	9	7	0
50	**2**	600	2242	601	4108	600	1593	600	1598	**603**	**154**	3528	0	15	10	0
	4	600	1638	600	1670	611	789	603	809	**622**	**118**	624	0	0	0	0
	6	600	3133	600	2978	604	1490	601	1519	**605**	**98**	200	1	0	0	0
	8	600	1682	600	1522	604	761	603	775	**605**	**110**	19	5	0	0	0

Table 11

Average results (in terms of time, gap and number of nodes in the branch-and-cut tree) obtained for each of the five methods over 10 *instances of* D_2*.* (BC) *corresponds to the branch-and-cut algorithm presented in Section 4. Bold results are the best in terms of gap and time. A gap of* 0% *corresponds to an optimal solution over each of the 10 instances.*

n	**K**	**Time (s) and Gap (%)**										**Nodes**				
		(F_{nc1})		(F_{nc2})		(F_{er})		(F_{ext})		(BC)		(F_{nc1})	(F_{nc2})	(F_{er})	(F_{ext})	(BC)
		time	gap	time	gap	time	gap	time	gap	time	gap					
20	**2**	**0**	**0**	1	**0**	4	**0**	4	**0**	4	**0**	8	**0**	19	18	15
	4	4	**0**	**2**	**0**	7	**0**	10	**0**	9	**0**	1408	102	143	142	124
	6	130	**0**	**3**	**0**	19	**0**	17	**0**	22	**0**	36589	456	681	631	575
	8	600	19	**3**	**0**	31	**0**	34	**0**	47	**0**	62536	1163	2017	1486	1438
30	**2**	**0**	**0**	3	**0**	42	**0**	43	**0**	35	**0**	50	0	33	32	29
	4	28	**0**	**14**	**0**	133	**0**	146	**0**	128	**0**	3769	342	387	402	272
	6	600	7	**21**	**0**	305	**0**	314	1	496	1	16951	1527	2000	1724	934
	8	601	25	**414**	**3**	564	5	564	6	602	7	7771	4806	4506	3970	1093
40	**2**	**1**	**0**	13	**0**	245	**0**	199	**0**	202	**0**	72	35	48	50	32
	4	407	**0**	**52**	**0**	542	3	549	3	605	5	8274	953	405	333	15
	6	601	14	**462**	**4**	600	9	600	9	606	12	3946	2906	428	355	18
	8	600	27	**600**	**9**	600	14	600	14	604	18	1625	2451	428	419	20
50	**2**	**3**	**0**	34	**0**	486	1	499	1	603	1	92	64	35	37	7
	4	600	5	**258**	**0**	600	5	600	5	609	6	3829	1771	21	22	4
	6	600	15	**600**	**6**	600	10	600	10	608	10	1145	1881	21	18	5
	8	600	29	**600**	**13**	600	18	600	17	610	15	453	1662	18	19	4

Table 12

Average results (in terms of time, gap and number of nodes in the branch-and-cut tree) obtained for each of the five methods over 10 *instances of* D_3*.* (BC) *corresponds to the branch-and-cut algorithm presented in Section 4. Bold results are the best in terms of gap and time. A gap of* 0% *corresponds to an optimal solution over each of the 10 instances.*

Instance	**n**	**K**	**Time (s) and Gap (%)**										**Nodes**				
			(F_{nc1})		(F_{nc2})		(F_{er})		(F_{ext})		(BC)		(F_{nc1})	(F_{nc2})	(F_{er})	(F_{ext})	(BC)
			time	gap	time	gap	time	gap	time	gap	time	gap					
bayg	**29**	**2**	19	**0**	468	**0**	3251	**0**	112	**0**	**3**	**0**	1534	4287	35141	265	0
		4	3600	10	3602	19	3600	1	3600	1	**156**	**0**	210026	22604	41339	34090	57
		6	3603	16	3600	9	230	**0**	304	**0**	**14**	**0**	230472	42730	6442	4875	0
		8	3601	16	1966	**0**	40	**0**	109	**0**	**9**	**0**	215260	33339	1541	1859	0
swiss	**42**	**2**	20	**0**	1024	**0**	3601	6	3037	**0**	**19**	**0**	246	548	2516	1309	0
		4	3600	7	3600	11	3603	8	3601	7	**19**	**0**	54962	2821	13652	3534	0
		6	3601	11	3601	7	3602	8	3603	6	**11**	**0**	66849	3673	30882	11899	0
		8	3604	12	3601	3	3601	6	3601	2	**9**	**0**	88964	4932	59914	19128	0
att	**48**	**2**	**26**	**0**	1231	**0**	3601	5	1388	**0**	35	**0**	167	140	1385	197	0
		4	552	**0**	3503	**0**	3601	4	2285	**0**	**18**	**0**	5035	2596	4142	1940	0
		6	3602	9	3600	6	3606	7	3602	6	**19**	**0**	33152	2645	19753	5477	0
		8	3601	13	3600	2	3604	5	3605	3	**14**	**0**	25983	4174	46173	10428	0
berlin	**52**	**2**	179	**0**	3600	18	3601	24	3600	21	**128**	**0**	1323	278	787	79	0
		4	3602	17	3602	20	3601	21	3601	19	**75**	**0**	23673	743	1909	590	0
		6	3602	32	3600	25	3604	27	3601	24	**1679**	**0**	20848	1184	7629	2093	28
		8	3613	31	3600	20	3608	24	3602	21	**45**	**0**	33560	1280	18456	4282	0
st	**70**	**2**	629	**0**	3600	12	197	**0**	308	**0**	**86**	**0**	859	0	0	0	0
		4	3600	4	3600	7	3600	3	3600	2	**3606**	**0**	2941	31	79	49	0
		6	3601	13	3601	10	3602	5	3603	4	**566**	**0**	4342	142	694	206	5
		8	3602	16	3605	11	3600	8	3603	7	**3604**	**3**	3557	319	1300	350	8
pr	**76**	**2**	392	**0**	3600	15	3600	2	3600	2	**145**	**0**	141	0	12	1	0
		4	3601	22	3600	21	3600	13	3600	12	**3603**	**1**	2803	2	56	3	0
		6	3601	25	3600	18	3602	11	3600	11	**3604**	**1**	2087	13	198	17	0
		8	3601	23	3600	13	3604	9	3600	8	**96**	**0**	1908	26	358	32	0
eil	**101**	**2**	3601	9	3601	19	3604	4	3602	4	**3600**	**1**	1287	0	0	0	0
		4	3600	19	3600	20	3600	10	3601	9	**3602**	**6**	294	0	3	0	0
		6	3600	35	3601	32	3600	24	3600	24	**3600**	**18**	57	0	7	2	0
		8	3600	34	3616	30	3600	24	3600	23	**3628**	**16**	41	0	39	5	0
bier	**127**	**2**	3601	25	3601	37	3601	32	3606	31	**3603**	**15**	869	0	0	0	0
		4	3618	54	3601	54	3601	49	3603	48	**3676**	**31**	10	0	2	0	0
		6	3600	69	3601	69	3601	65	3664	64	**3602**	**53**	13	0	2	0	0
		8	3600	65	3601	63	3600	59	3601	58	**3600**	**47**	7	0	0	0	0

Table 13

Average results (in terms of time, gap and number of nodes in the branch-and-cut tree) obtained for each of the five methods over 8 instances of the TSPLIB. (BC) corresponds to the branch-and-cut algorithm presented in Section 4. Bold results are the best in terms of gap and time. A gap of 0% corresponds to an optimal solution.

K	**Time (s) and Gap (%)**										**Nodes**				
	(F_{nc1})		(F_{nc2})		(F_{er})		(F_{ext})		(BC)		(F_{nc1})	(F_{nc2})	(F_{er})	(F_{ext})	(BC)
	time	gap	time	gap	time	gap	time	gap	time	gap					
2	3031	4	3600	19	3681	2	3629	2	**1143**	**0**	8211	0	0	0	0
4	3601	13	3608	14	3601	2	3600	2	**3629**	**2**	1281	0	5	1	0
6	3600	12	3600	12	1993	**0**	1913	**0**	**1869**	**0**	166	0	6	0	0
8	3600	12	3601	12	1438	**0**	**1240**	**0**	1589	**0**	11	0	1	0	0

Table 14

Results (in terms of time, gap and number of nodes in the branch-and-cut tree) obtained for each of the method over 6 instances of 100 nodes from the spin-glass data set. (BC) *corresponds to the branch-and-cut algorithm presented in Section 4. Bold results are the best in terms of gap and time. A gap of* 0% *corresponds to an optimal solution over each of the 6 instances.*

References

[1] Z. Ales, Extraction et partitionnement pour la recherche de régularités : Application à l'analyse de dialogues, Ph.D. thesis, INSA de Rouen, 2014.

[2] Z. Ales and A. Knippel, *An extended edge-representative formulation for the k-partitioning problem*, Electron. Notes Discr. Math. **52** (2016), 333–342.

[3] Z. Ales, A. Knippel, and A. Pauchet, *On the polyhedron of the k-partitioning problem with representative variables*, Tech. report, LMI/LITIS, INSA de Rouen, 2014.

[4] Z. Ales, A. Knippel, and A. Pauchet, *Polyhedral combinatorics of the k-partitioning problem with representative variables*, Discr. Appl. Math. **211** (2016), 1–14.

[5] Z. Ales, A. Pauchet, and A. Knippel, *Extraction de motifs dialogiques bidimensionnels*, Revue d'intelligence artificielle **29** (2015), 655–684.

[6] Z. Ales, A. Pauchet, and A. Knippel, *Extraction and clustering of two-dimensional dialogue patterns*, Int. J. Artificial Intelligence Tools **27** (2018), 1850001.

[7] F. Barahona and A. Mahjoub, *On the cut polytope*, Math. Program. **36** (1986), 157–173.

[8] P. Berkhin, "*A survey of clustering data mining techniques*," *Grouping multidimensional data*, T.M. Kogan J., Nicholas C. (ed.), Springer, Pavel Berkhin, Accrue Software, 1045 Forest Knoll Dr., San Jose, CA, 2006, pp. 25–71.

[9] P. Bonami, V. Nguyen, M. Klein, and M. Minoux, *On the solution of a graph partitioning problem under capacity constraints*, Combinatorial Optimization, Vol. 7422 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, pp. 285–296.

[10] M. Boulle, *Compact mathematical formulation for graph partitioning*, Optim. Eng. **5** (2004), 315–333.

[11] M. Campêlo, V. Campos, and R. Corrêa, *On the asymmetric representatives formulation for the vertex coloring problem*, Discr. Appl. Math. **156** (2008), 1097–1111.

[12] S. Chopra and M. Rao, *The partition problem*, Math. Program. **59** (1993), 87–115.

[13] S. Chopra and M. Rao, *Facets of the k-partition polytope*, Discr. Appl. Math. **61** (1995), 27–48.

[14] G. Dahl and T. Flatberg, *An integer programming approach to image segmentation and reconstruction problems*, Geometric Modelling, Numer. Simulation, Optim. (2007), 475–496.

[15] M. Deza, M. Grötschel, and M. Laurent, *Clique-web facets for multicut polytopes*, Math. Oper. Res. **17** (1992), 981–1000.

[16] N. Fan and P. Pardalos, *Linear and quadratic programming approaches for the general graph partitioning problem*, J. Global Optim. **48** (2010), 57–71.

[17] N. Fan, Q. Zheng, and P. Pardalos, *Robust optimization of graph partitioning involving interval uncertainty*, Theor. Comput. Sci. **447** (2012), 53–61.

[18] C. Ferreira, A. Martin, C. De Souza, R. Weismantel, and L. Wolsey, *Formulations and valid inequalities for the node capacitated graph partitioning problem*, Math. Program. **74** (1996), 247–266.

[19] C. Ferreira, A. Martin, C. De Souza, R. Weismantel, and L. Wolsey, *The node capacitated graph partitioning problem: A computational study*, Math. Program. **81** (1998), 229–256.

[20] M. Garey, D. Johnson, and L. Stockmeyer, *Some simplified NP-complete graph problems*, Theor. Comput. Sci. **1** (1976), 237–267.

[21] O. Goldschmidt and D. Hochbaum, *A polynomial algorithm for the k-cut problem for fixed k*, Math. Oper. Res. **19** (1994), 24–37.

[22] M. Grötschel and Y. Wakabayashi, *A cutting plane algorithm for a clustering problem*, Math. Program. **45** (1989), 59–96.

[23] M. Grötschel and Y. Wakabayashi, *Facets of the clique partitioning polytope*, Math. Program. **47** (1990), 367–387.

[24] P. Hansen and B. Jaumard, *Cluster analysis and mathematical programming*, Math. Program. **79** (1997), 191–215.

[25] B. Hendrickson and R. Leland, *An improved spectral graph partitioning algorithm for mapping parallel computations*, SIAM J. Sci. Comput. **16** (1995), 452–469.

[26] A. Jain, *Data clustering: 50 years beyond k-means*, Pattern Recognition Lett. **31** (2010), 651–666.

[27] E. Johnson, A. Mehrotra, and G. Nemhauser, *Min-cut clustering*, Math. Program. **62** (1993), 133–151.

[28] V. Kaibel, M. Peinhardt, and M. Pfetsch, *Orbitopal fixing*, Discr. Optim. **8** (2011), 595–610.

[29] B. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, Bell System Tech. J. **49** (1970), 291–307.

[30] S. Khuller and B. Saha, *On finding dense subgraphs*, Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I, Springer, 2009, pp. 597–608.

[31] M. Labbé and F. Özsoy, *Size-constrained graph partitioning polytopes*, Discr. Math. **310** (2010), 3473–3493.

[32] H. Li and C. Shen, *Interactive color image segmentation with linear programming*, Machine Vision Appl. **21** (2010), 403–412.

[33] F. Liers, Contributions to determining exact ground-states of Ising spin-glasses and to their physics, Ph.D. thesis, Universität zu Köln, 2004.

[34] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi, *Computing exact ground states of hard Ising spin glass problems by branch-and-cut*, New Optimization Algorithms in Physics, John Wiley & Sons, Ltd, 2004, pp. 47–69.

[35] A. Lisser and F. Rendl, *Graph partitioning using linear and semidefinite programming*, Math. Program. **95** (2003), 91–101.

[36] J. Mitchell, *Branch-and-cut for the k-way equipartition problem*, Tech. report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, 2001.

[37] M. Oosten, J. Rutten, and F. Spieksma, *The clique partitioning problem: Facets and patching facets*, Networks **38** (2001), 209–226.

[38] G. Reinelt, *TSPLIB - A traveling salesman problem library*, ORSA J. Computing (1991), 376–384.

[39] M. Sørensen, *b-tree facets for the simple graph partitioning polytope*, J. Combinatorial Optim. **8** (2004), 151–170.

[40] M. Sørensen, *Facet-defining inequalities for the simple graph partitioning polytope*, Discr. Optim. **4** (2007), 221–231.

[41] C. Walshaw, M. Cross, and M. Everett, *Parallel dynamic graph partitioning for adaptive unstructured meshes*, J. Parallel Distrib. Comput. **47** (1997), 102–108.

Appendix

Bound 6.1 *The optimal value of the linear relaxation of* (F_{nc2}) *is less than or equal to* $\min\limits_{j\in\{2,\dots,K\}} \min\limits_{i<j} \frac{w_{ij}}{2^{K-j+1}}$.

Proof. To prove this bound, we exhibit for all $j \in \{2,\dots,K\}$ and all $i \in \{1,\dots,j-1\}$, a feasible solution of the linear relaxation in which x_{ij} is equal to 2^{j-K-1} and all the remaining x variables are null.

To obtain such a solution, we fix the value of the z variables as follows:

- $z_a^a = 1 \quad \forall a \leq j-1$;
- $z_j^i = 2^{j-K-1}$;
- $z_j^j = 1 - z_j^i$;
- $z_a^j = z_j^i \quad \forall a > j$;
- $z_a^a = 1 - 2^{a-K-2} \quad \forall a \in \{j+1,\dots,K+1\}$;
- $z_b^a = 2^{a-K-2} \quad \forall a \in \{j+1,\dots,K+1\} \quad \forall b \in \{a+1,\dots,n\}$;
- $z_a^b = 0$ otherwise.

Table 15 represents the value of these coefficients when i is equal to 1 and j is strictly less than K.

Node	Cluster								
	1	…	$j-1$	j	$j+1$	$j+2$	…	K	$K+1$
1	1								
$\vdots$		$\ddots$							
$j-1$			1						
j	2^{j-K-1}			$1-2^{j-K-1}$					
$j+1$				2^{j-K-1}	$1-2^{j-K-1}$				
$j+2$				$\vdots$	2^{j-K-1}	$1-2^{j-K}$			
$\vdots$				$\vdots$	$\vdots$	2^{j-K}	$\ddots$		
K				$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\frac{3}{4}$	
$K+1$				$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\frac{1}{4}$	$\frac{1}{2}$
$\vdots$				$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n				2^{j-K-1}	2^{j-K-1}	2^{j-K}	…	$\frac{1}{4}$	$\frac{1}{2}$

Table 15

For a given $j < K$, representation of the variables z which are not null in a feasible solution of cost $2^{j-K-1}w_{1j}$ of the linear relaxation of formulation (F_{nc2}).

We let the reader check that these solutions are feasible for the linear relaxation of (F_{nc2}). For instance, the fact that the sum of each line of Table 15 is equal to 1 ensures that Equations (16) are satisfied. Similarly, Equation (21) is satisfied as the sum of the diagonal variables in Table 15 is equal to K. □

Bound 6.2 *If $K \in \{3, \dots, n\}$, then the optimal value of the linear relaxation of (F_{nc1}) is less than or equal to $\min\limits_{j\in\{2,\dots,K-1\}} \min\limits_{i<j} \frac{w_{ij}}{2^{K-j}}$.*

Proof. To prove this bound, we exhibit for all $j \in \{2, \dots, K-1\}$ and all $i \in \{1, \dots, j-1\}$, a feasible solution of the linear relaxation in which x_{ij} is equal to 2^{j-K} and all the remaining x variables are null.

To obtain such a solution, we fix the value of the z variables as follows:

- $z_a^a = 1 \quad \forall a \leq j-1$;
- $z_j^i = 2^{j-K}$;
- $z_j^j = 1 - z_j^i$;
- $z_a^j = z_j^i \quad \forall a > j$;
- $z_a^a = 1 - 2^{a-K-1} \quad \forall a \in \{j+1, \dots, K\}$;
- $z_b^a = 2^{a-K-1} \quad \forall a \in \{j+1, \dots, K\} \quad \forall b \in \{a+1, \dots, n\}$;
- $z_a^b = 0$ otherwise.

Table 16 represents the value of these coefficients when i is equal to 1 and j is strictly less than $K-1$.

□

Node	**Cluster**								
	1	...	$j-1$	j	$j+1$	$j+2$	...	$K-1$	K
1	1								
$\vdots$		$\ddots$							
$j-1$			1						
j	2^{j-K}			$1-2^{j-K}$					
$j+1$				2^{j-K}	$1-2^{j-K}$				
$j+2$				$\vdots$	2^{j-K}	$1-2^{j-K+1}$			
$\vdots$				$\vdots$	$\vdots$	2^{j-K+1}	$\ddots$		
K				$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\frac{3}{4}$	
$K+1$				$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\frac{1}{4}$	$\frac{1}{2}$
$\vdots$				$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n				2^{j-K}	2^{j-K}	2^{j-K+1}	...	$\frac{1}{4}$	$\frac{1}{2}$

Table 16

For a given $j < K-1$, representation of the variables z which are not null in a feasible solution of cost $2^{j-K}w_{1j}$ of the linear relaxation of formulation (F_{nc1}).

Bound 6.3 *If K is equal to 2, then the optimal value of the linear relaxation of (F_{nc1}) is in the interval $[\min\limits_{i,j\in V} w_{ij}\frac{n-1}{2}, \frac{1}{2}\sum\limits_{i=2}^{n} w_{1i}]$.*

Proof. We consider a solution in which:

- $x_{1i} = 0.5$ for all $i \in V$;
- $x_{ij} = 0$ otherwise;
- $z_1^1 = 1$;
- $z_i^j = 0.5$ otherwise.

This solution is feasible, hence the optimal value of the linear relaxation is less than or equal to $\frac{1}{2}\sum\limits_{i=2}^{n} w_{1i}$.

To prove that $\min\limits_{i,j\in V} w_{ij}\frac{n-1}{2}$ is a lower bound, we show that $\sum\limits_{ij\in E} x_{ij}$ is necessarily greater than or equal to $\frac{n-1}{2}$.

From Equation (15) and the fact that z_1^1 is equal to 1, we deduce that

$$x_{ij} \geq z_i^2 + z_j^2 - 1 \quad \forall ij \in E, \tag{51}$$

and

$$x_{1i} \geq z_i^1 \quad \forall i \in V\backslash\{1\}. \tag{52}$$

Let $S \subset V$ be the set of nodes i which satisfy $z_i^2 > 0.5$. We deduce from (51) and (52) that

$$\sum_{ij\in E} x_{ij} \geq \sum_{i\geq 2} z_i^1 + \sum_{i,j\in S} z_i^2 + z_j^2 - 1. \tag{53}$$

We know from (16) that

$$\begin{aligned} \tfrac{n-1}{2} &= \textstyle\sum_{i\geq 2}(z_i^1 + z_i^2 - \tfrac{1}{2}) \\ &\leq \textstyle\sum_{i\geq 2} z_i^1 + \sum_{i\in S}(z_i^2 - \tfrac{1}{2}) \end{aligned} \tag{54}$$

The lower bound is obtained from (53) and (54).

□