# A branch-and-cut algorithm for the maximum k-balanced subgraph of a signed graph 

Rosa Figueiredo, Yuri Y. Frota, Martine Labbé

## - To cite this version:

Rosa Figueiredo, Yuri Y. Frota, Martine Labbé. A branch-and-cut algorithm for the maximum k-balanced subgraph of a signed graph. Discrete Applied Mathematics, 2018, 261, pp.164-185. 10.1016/j.dam.2018.11.022 . hal-01937015

HAL Id: hal-01937015
https://inria.hal.science/hal-01937015
Submitted on 27 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A branch-and-cut algorithm for the maximum $k$-balanced subgraph of a signed graph 

Rosa Figueiredo ${ }^{\text {a,* }}$, Yuri Frota ${ }^{\text {b }}$, Martine Labbé ${ }^{\text {c }}$<br>${ }^{a}$ Laboratoire Informatique d'Avignon, University of Avignon<br>84911 Avignon cedex 9, France.<br>rosa.figueiredo@univ-avignon.fr<br>${ }^{b}$ Department of Computer Science, Fluminense Federal University<br>24210-240 Niterói-RJ, Brazil. yuri@ic.uff.br<br>${ }^{c}$ Université Libre de Bruxelles, Département d'Informatique<br>CP 210/01 B-1050 Brussels, Belgium.<br>mlabbe@ulb.ac.be


#### Abstract

We are interested in the solution of the maximum $k$-balanced subgraph problem. Let $G=(V, E, s)$ be a signed graph and $k$ a positive scalar. A signed graph is $k$-balanced if $V$ can be partitioned into at most $k$ sets in such a way that positive edges are found only within the sets and negative edges go between sets. The maximum $k$-balanced subgraph problem is the problem of finding a subgraph of $G$ that is $k$-balanced and maximum according to the number of vertices. This problem has applications in clustering problems appearing in collaborative $v s$ conflicting environments. The particular case $k=2$ yields the problem of finding a maximum balanced subgraph in a signed graph and its exact solution has been addressed before in the literature. In this paper, we provide a representatives formulation for the general problem and present a partial description of the associated polytope, including the introduction of strengthening families of valid inequalities. A branch-and-cut algorithm is described for finding an optimal solution to the problem. An ILS metaheuristic is implemented for providing primal bounds for this exact method and a branching rule strategy is proposed for the representatives formulation. Computational experiments, carried out over a set of random instances and on a set of instances from an application, show the effectiveness of the valid inequalities and strategies adopted in this work.


Keywords: Signed graph; Balanced graph; Graph partition; Integer programming; Social networks.

[^0]
## 1. Introduction

Let $G=(V, E)$ be an undirected graph where $V=\{1,2, \ldots, n\}$ is a set of vertices and $E$ is a set of edges connecting pairs of vertices. 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. Let $G=(V, E, s)$ denote a signed graph. We assume that a signed graph has no parallel edges. An edge $e \in E$ is called negative if $s(e)=-$ and positive if $s(e)=+$. Let $E^{-}$and $E^{+}$denote, respectively, the sets of negative and positive edges in a signed graph. Notice that, according to the definitions above, $E=E^{-} \cup E^{+}$and $E^{-} \cap E^{+}=\emptyset$. For a vertex set $S \subseteq V$, let $E[S]=\{(i, j) \in E \mid i, j \in S\}$ denote the subset of edges induced by $S$. A signed graph $G$ is balanced if its vertex set can be partitioned into sets $W$ (possibly empty) and $V \backslash W$ in such a way that $E[W] \cup E[V \backslash W]=E^{+}$. Let $k$ be a given parameter satisfying $k \in\{1, \ldots, n\}$. A signed graph $G$ is $k$-balanced if its vertex set $V$ can be partitioned into sets $N_{1}, N_{2}, \ldots, N_{l}$, with $l \leq k$, in such a way that $\cup_{1 \leq i \leq l} E\left[N_{i}\right]=E^{+}$. Such a partition is called a l-partition in what follows. A signed graph is clusterable if it is $n$-balanced (see Figure 1). A clusterable signed graph is called a colorable signed graph in [13]. The problem that is studied herein can be stated as follows.

Definition 1.1 ( $k$-MBS problem). Let $G=(V, E, s)$ be a signed graph and $k$ be a scalar value satisfying $k \in\{1, \ldots, n\}$. The maximum $k$-balanced subgraph problem is the problem of finding an induced subgraph $H=\left(V^{\prime}, E\left[V^{\prime}\right], s\right)$ of $G$ such that $H$ is $k$-balanced and maximizes the cardinality of the vertex set $V^{\prime}$. Let us denote this maximal value by $v(G, k)$.

The particular case $k=2$ yields the problem of finding a maximum balanced subgraph in a signed graph $[8,24,26]$. The $k$-MBS problem is closely related to a clustering problem defined over a weighted signed graph $G$ : the correlation clustering problem [7]. An optimal solution for the $k$-MBS problem defined on $G$ is a subgraph of $G$ that admits a correlation clustering solution with no disagreements.

Signed graphs were introduced by Heider in [34] with the purpose of describing sentiment relations between people pertaining to the same social group and to provide a systematic statement of social balance theory. Cartwright et al. [12] formalized Heider's theory, stating that a balanced signed graph could be partitioned into two mutually antagonistic subgroups of vertices each having internal solidarity. The definition of a balanced signed graph was extended to the definition of a clusterable signed graph in [17]. ${ }^{1}$ We refer the reader to [51] for a bibliography of signed graphs.

In the last decades, signed graphs have shown to be a very attractive discrete structure for social network researchers. Analyzing and mining social networks can help gain a better understanding of human behavior in social groups. In [19,

[^1]

Figure 1: (a) A balanced signed graph: $W=\{1,2,6\}$ and $V \backslash W=\{3,4,5\}$. (b) A $k$-balanced signed graph for any $k \in\{3,4,5,6\}$. Sets $N_{1}=\{1,2\}, N_{2}=\{3,4,5\}$ and $N_{3}=\{6\}$ define a 3 -partition of this graph. (c) A non-clusterable signed graph.
$20,22,27,41,43]$ the solution of clustering problems defined on signed graphs was used as a tool for the evaluation of structural balance in a social network. Balancing on signed graphs was used in [37, 38, 39, 40] to discuss properties and concepts in the context of group decision making. Applications of balancing problems defined on signed graphs also appear in soft game theory [36], in system biology [16], in portfolio analysis in risk management [24, 33], in voting behavior [4] and in the detection of embedded matrix structures [24].

The $k$-MBS problem has applications in clustering problems appearing in collaborative vs. conflicting environments. Consider a social group (for example, a political decision group). In this context, each vertex in the signed graph represents a person in this social group while an edge represents a sentiment relation (mutual liking or disliking, friendship or enmity, cooperation or defection, interaction or avoidance) between a pair of vertices and determined by past observations. An optimal solution of the $k$-MBS problem provides us with an efficient way to define $l \leq k$ teams (groups with political allies): positive relations are allowed only within teams, negative relations link different teams while the number of elements involved is maximized. Additional restrictions can be taken into account, such as size constraints for the teams or a fixed number of teams. A weighted version of the $k$-MBS problem can be defined whenever weights are used to graduate positive and negative relations. Balancing on signed graphs is used in [33] to define a risk-limiting strategy for portfolio design. In this context, a security is represented by a vertex in the signed graph while the correlation between securities is represented by the set of signed edges. The efficient solu-
tion of the $k$-MBS problem defined over a signed graph representing a big set of securities gives an efficient way to define a limited-risk portfolio. We refer the reader to [24] for a more detailed discussion about applications of the $k$-MBS problem.

Solution approaches have been proposed in the literature for the 2-MBS problem. In [32], Harary presented a simple linear procedure to detect whether a signed graph is 2-balanced. Descriptions of the polytope associated with the 2-MBS problem were given in [8, 26]. A greedy heuristic approach was proposed in [31] for this particular case and is able to find an optimal solution whenever $G$ is a balanced signed graph. Recently, Figueiredo et al. [24] contributed to the efficient solution of the 2-MBS problem by developing a pre-processing routine, an efficient GRASP metaheuristic, and improved versions of the greedy heuristic proposed in [31]. A branch-and-cut algorithm was proposed in [23, 26] to the exact solution of the 2 -MBS problem. Computational approaches (including optimization approaches) have been described in the literature over the past years for the solution of other balancing problems defined on signed graphs [2, $15,18,19,20,21,27,50]$. Most of them have been used to analyze and mine social networks. To the best of our knowledge, the general case of the $k$-MBS problem has never been treated in the literature before.

In this paper, we contribute to the efficient solution of the $k$-MBS problem. We provide an integer linear programming formulation to the problem strengthened by the introduction of families of valid inequalities of the associated polytope. Based on these results, a branch-and-cut algorithm is developed to the problem. A primal heuristic is implemented for feeding the exact method with primal bounds and a constraint branching rule is proposed. Preliminary results were previously reported in a four-page conference paper [25].

We next give some notations and definitions to be used throughout the paper. Let $G=(V, E)$ be an undirected graph. For a vertex set $S \subseteq V$, we define $N(S)=\{j \in V \mid(i, j) \in E, i \in S\}$. A set $K \subseteq V$ is a clique in $G$ if each pair of vertices $i, j \in K$ is joined by an edge $(i, j) \in E$. A set $I \subseteq V$ is a stable set in $G$ if no pair of vertices $i, j \in I$ is joined by an edge in $E$; i.e. $(i, j) \notin E$. Let $\alpha(G)$ denote the cardinality of a maximum stable set in $G$. We represent a cycle by its vertex set $C \subseteq V$. A hole is a chordless cycle of length at least four. An odd hole is a hole with an odd number of vertices. Now, consider a function $s: E \rightarrow\{+,-\}$ and let $G=(V, E, s)$ denote a signed graph. Definitions and notations for an undirected graph $G=(V, E)$ also apply to the signed graph $G=(V, E, s)$. We define $G^{-}=\left(V, E^{-}\right)$and $G^{+}=\left(V, E^{+}\right)$. For a vertex set $S \subseteq V$, we define $N^{-}(S)=\left\{j \in V \mid(i, j) \in E^{-}, i \in S\right\}$, $N^{+}(S)=\left\{j \in V \mid(i, j) \in E^{+}, i \in S\right\}$ and $\bar{N}^{-}(S)=V \backslash\left(N^{-}(S) \cup S\right)$. A set $K \subseteq V$ is called a negative (positive) clique if each pair of vertices in $K$ is joined by a negative (positive) edge. A positive (negative) hole is a hole with only positive (negative) edges.

## 2. Complexity and integer formulation

Let $G=(V, E, s)$ be a signed graph and $k$ a positive scalar satisfying $k \in$ $\{1, \ldots, n\}$. The $k$-MBS problem is NP-hard. The particular case $k=2$ is proven to be NP-hard in [9]. Indeed, the $k$-MBS problem is the optimization version of an NP-complete decision problem, namely, the induced subgraph with property $\pi$ (property $\pi$ must hold for arbitrarily large graphs, do not hold for all graphs and be hereditary, i.e., it is valid for all induced subgraphs of $G$ whenever it is valid for $G)([29], \mathrm{pp}$. 195). For the $k$-MBS problem, property $\pi$ is " $H$ is $k$-balanced".

Next, we see that the optimal solution of the independent set problem defined over $G^{-}$provides a lower bound for the $k$-MBS problem.

Proposition 2.1. $\alpha\left(G^{-}\right)=v(G, 1) \leq v(G, 2) \leq \ldots \leq v(G, n)$.
Proof. Consider a signed graph $G$. The feasible solution space of the $(k-1)$ MBS problem is contained in the feasible solution space of the $k$-MBS problem. Then, $v(G, k-1) \leq v(G, k)$, for all, $2 \leq k \leq n$. It is not difficult to check that each feasible solution for the 1-MBS problem defined over $G$ is a feasible solution for the stable set problem defined over $G^{-}$, and vice-versa. A feasible solution for the 1-MBS problem is by definition a 1-balanced subgraph $H=\left(V^{H}, E^{H}\right)$ of $G$ which implies $E\left[V^{H}\right]=E^{H} \subseteq E^{+}$and $V^{H}$ is an independent set in $G^{-}$. A feasible solution for the independent set problem defined over $G^{-}=\left(V, E^{-}\right)$ is a vertex set $I \subseteq V$ such that for all $i, j \in I,(i, j) \notin E^{-}$. That implies $H=(I, E[I])$ is a 1-balanced subgraph of $G$.

In this work, we propose a discrete linear formulation for the $k$-MBS problem; more precisely, we propose a representatives formulation for this problem. Integer linear formulations have already been presented for correlationclustering problems [18, 27] and for the 2-MBS problem [26]. Representatives formulations have been successfully applied to the solution of graph clustering problems $[3,6,10,11,14,27,28]$.

We define the arc sets $A=\left\{(i, j) \mid i \in V, j \in \bar{N}^{-}(\{i\}), i<j\right\}$ and $A^{0}=$ $A \cup\{(i, i) \mid i \in V\}$. An arc $(i, j) \in A^{0}$ indicates that vertex $i$ can represent vertex $j$, or equivalently that vertex $j$ can be represented by vertex $i$. Let $D(i)=\left\{j \in V \mid(i, j) \in A^{0}\right\}$ denote the set of all vertices that can be represented by $i$ and let $O(j)=\left\{i \in V \mid(i, j) \in A^{0}\right\}$ denote the set of all vertices that can represent $j$. Additionally, let $S_{i}$ denote a vertex set such that: $S_{i} \subseteq D(i)$ and $i \in S_{i}$. We say that $i$ is the representative vertex of $S_{i}$.

A feasible solution in our formulation defines a $k$-balanced subgraph of $G$. Let $n=|V|$ and $m=|A|$. To describe this feasible set, we use binary decision variables $x \in\{0,1\}^{n+m}$ that define a $l$-partition, $l \leq k$, of a $k$-balanced subgraph. For each vertex $i \in V$,
$x_{i}^{i}= \begin{cases}1 & \text { if } i \text { is a representative vertex, i.e., } \\ \text { a set } S_{i} \text { belongs to the } l \text {-partition }, \\ 0 & \text { otherwise. }\end{cases}$

Also, for each $\operatorname{arc}(i, j) \in A$, we define
$x_{j}^{i}= \begin{cases}1 & \text { if vertex } j \text { is represented by vertex } i, \text { i.e. } \\ j \text { belongs to a set } S_{i} \text { in the } l \text {-partition }, \\ 0 & \text { otherwise }\end{cases}$
The formulation follows.

$$
\begin{align*}
& \text { Maximize } \sum_{(i, j) \in A^{0}} x_{j}^{i}  \tag{1}\\
& \text { subject to } \sum_{i \in O(j)} x_{j}^{i} \leq 1, \quad \forall j \in V \text {, }  \tag{2}\\
& \sum_{i \in V} x_{i}^{i} \leq k,  \tag{3}\\
& x_{j}^{i} \leq x_{i}^{i}, \quad \forall(i, j) \in A,  \tag{4}\\
& x_{i}^{p}+x_{j}^{p} \leq x_{p}^{p}, \quad \forall(i, j) \in E^{-}, \forall p \in O(i) \cap O(j),  \tag{5}\\
& x_{i}^{p}+x_{j}^{q} \leq 1, \quad \forall(i, j) \in E^{+}, \forall p \in O(i), \forall q \in O(j) \backslash\{p\},  \tag{6}\\
& x_{i}^{i} \in\{0,1\}, \quad \forall i \in V,  \tag{7}\\
& x_{j}^{i} \in\{0,1\}, \quad \forall(i, j) \in A . \tag{8}
\end{align*}
$$

Constraints (2) establish that vertex $j$ must be represented by at most one vertex, i.e., vertex $j$ must belong to at most one cluster in the $l$-partition. The total number of representative vertices is limited to $k$ by constraint (3), i.e., the total number of clusters in the $l$-partition is limited to $k$. Constraints (4) forbid vertex $j$ to be represented by vertex $i$ unless $i$ is a representative vertex. Consider a negative edge $(i, j) \in E^{-}$. Constraints (5), written for $(i, j)$, ensure that vertices $i$ and $j$ cannot be represented by a same vertex, i.e., $i$ and $j$ cannot belong to a same cluster in the $l$-partition. Consider a positive edge $(i, j) \in E^{+}$. Constraints (6), written for $(i, j)$, ensure that vertices $i$ and $j$ are represented by the same vertex whenever both $i$ and $j$ belong to the feasible solution, i.e., in that case vertices $i$ and $j$ belong to the same cluster in the $l$-partition. Constraints (7) and (8) impose binary restrictions to the variables. Finally, the objective function (1) looks for a maximum subgraph. Let us refer to this formulation as $I P(G, k)$.

## 3. Polyhedral study

In this section, we assume that $G$ is a connected graph and that $k \geq 2$. The $k$-MBS polytope $P_{G, k}$ associated with formulation $\operatorname{IP}(G, k)$ is defined here as

$$
P_{G, k}=\operatorname{conv}\left\{x \in \Re^{n+m} \mid x \text { satisfies }(2)-(8)\right\}
$$

Additional notations will be necessary before we can proceed. Consider a set of $l$ disjoint vertex sets $I=\left\{N_{1}, \ldots, N_{l}\right\}$. The incidence vector $x^{I} \in \Re^{n+m}$
is defined as: for each $i \in V, x^{I^{i}}{ }_{i}=1$ if $i$ is the representative vertex of $N_{t}$, for some $1 \leq t \leq l$, and $x^{I^{i}}{ }_{i}=0$ otherwise; for each $(i, j) \in A, x^{I^{i}}{ }_{j}=1$ if $j \in N_{t}$, $i$ is the representative vertex of $N_{t}$, for some $1 \leq t \leq l$, and $x^{I^{i}}{ }_{j}=0$ otherwise. Moreover, let $I_{i}=\{\{i\}\}$, for $i \in V$, and let $I_{i j}=\{\{i, j\}\}$, for $(i, j) \in A$.

Lemma 3.1. The polytope $P_{G, k}$ is full-dimensional, i.e., $\operatorname{dim}\left(P_{G, k}\right)=n+m$.
Proof. Since $P_{G, k}$ contains the null vector, it is sufficient to present other $n+$ $m$ affinely independent solutions $x \in \Re^{n+m}$ in $P_{G, k}$. Consider the following solutions: $x^{I_{i}}$, for each $i \in V$, and $x^{I_{i j}}$, for each $(i, j) \in A$. Clearly, these $n+m$ affinely independent solutions belong to $P_{G, k}$.

First, we investigate necessary and sufficient conditions for some inequalities in formulation $\operatorname{IP}(G, k)$ to define facets of $P_{G, k}$.

Lemma 3.2 (Trivial inequalities).
(a) For all $(i, j) \in A, x_{j}^{i} \geq 0$ defines a facet of $P_{G, k}$;
(b) for all $(i, j) \in A, x_{j}^{i} \leq 1$ does not define a facet of $P_{G, k}$;
(c) for all $i \in V, x_{i}^{i} \geq 0$ defines a facet of $P_{G, k}$ if and only if, for each vertex $j \in V,(i, j) \notin A ;$
(d) for all $i \in V, x_{i}^{i} \leq 1$ defines a facet of $P_{G, k}$ if and only if (i) for each vertex $j \in V,(j, i) \notin A$, and (ii) for each vertex $j \in V,(i, j) \notin E^{+}$.

Proof. The proof is straightforward.
The methodology that is used in the facet-defining proofs of this paper is stated in the following. Let $b^{T} x \leq \beta$ be a valid inequality to polytope $P_{G, k}$ and $F=\left\{x \in P_{G, k} \mid b^{T} x=\beta\right\}$ be the proper face of $P_{G, k}$ defined by this inequality. In order to prove that $F$ is a facet of $P_{G, k}$, we assume there is an inequality $a^{T} x \leq \alpha$, valid for $P_{G, k}$, such that $F \subseteq F_{a}=\left\{x \in P_{G, k} \mid a^{T} x=\alpha\right\}$ and show that the inequality defining $F_{a}$ can be written as a positive scalar multiple of the inequality defining $F$ (see Theorem 3.6 in [45]).

Lemma 3.3. Consider a vertex $p \in V$. Inequality (2) written for $p$, i.e.,

$$
\sum_{i \in O(p)} x_{p}^{i} \leq 1
$$

defines a facet of polytope $P_{G, k}$ if and only if, for each $i \in V \backslash\{p\},(i, p) \notin E^{+}$.

Proof. Suppose that $a_{p}^{p}=\alpha$ and let $i \in O(p) \backslash\{p\}$. Define the set of disjoint vertex sets $I_{1}=\{\{p\},\{i\}\}$. From the incidence vectors $x^{I_{1}}$ and $x^{I_{i p}}$ in $a^{T} x=\alpha$ we can conclude that $a_{p}^{i}=\alpha$. Consider an $\operatorname{arc}(p, j) \in A$. The solutions $x^{I_{p}}$ and $x^{I_{p j}}$ lead to $a_{j}^{p}=0$. Now, consider an arc $(i, j) \in A$ such that $i, j \neq p$ and
define the set of disjoint vertex sets $I_{2}=\{\{p\},\{i, j\}\}$. The incidence vectors $x^{I_{1}}$ and $x^{I_{2}}$ imply $a_{j}^{i}=0$. Consider a vertex $i \in V \backslash\{p\}$. From the solutions $x^{I_{p}}$ and $x^{I_{1}}$ we can conclude that $a_{i}^{i}=0$. Finally, we argue that the inequality is not facet defining if there is a vertex $q$ such that $(q, p) \in E^{+}$. In this case, the inequality is dominated by inequality $\sum_{i \in O(p)} x_{p}^{i}+x_{q}^{q} \leq 1$, if $q>p$, and by inequality $\sum_{i \in O(p) \backslash\{q\}} x_{p}^{i}+x_{q}^{q} \leq 1$, if $q<p$.
Lemma 3.4. Consider an $\operatorname{arc}(p, q) \in$ A. Inequality (4) written for $(p, q)$, i.e.,

$$
x_{q}^{p}-x_{p}^{p} \leq 0,
$$

defines a facet of polytope $P_{G, k}$ if and only if, for each $(p, j) \in A,(q, j) \notin E^{-}$.
Proof. The incidence vector $x^{I_{p q}}$ and the null vector in $a^{T} x=\alpha$ lead us to $a_{q}^{p}=-a_{p}^{p}$. Consider a vertex $i \in V \backslash\{p\}$. We conclude that $a_{i}^{i}=0$ from the null vector and the incidence vector $x^{I_{i}}$. Let $(i, j) \in A$ be an arc such that $i \neq p$. From $x^{I_{i j}}$ and the null vector we arrive to $a_{j}^{i}=-a_{i}^{i}=0$. Now consider an $\operatorname{arc}(p, j) \in A$, with $j \neq q$ and define the set $I_{2}=\{\{p, q, j\}\}$. Notice that, $x^{I_{2}} \in P_{G, k}$ since $(q, j) \notin E^{-}$. From solutions $x^{I_{p q}}$ and $x^{I_{2}}$ we conclude that $a_{j}^{p}=0$. Finally, assume that there is an arc $(p, j) \in A$ such that $(q, j) \in E^{-}$. In this case, the associated inequality is dominated by an inequality (5) written for edge $(q, j)$ and vertex $p$, i.e, $x_{q}^{p}+x_{j}^{p} \leq x_{p}^{p}$.

Constraints (6) can be replaced in formulation $\operatorname{IP}(G, k)$ by the stronger inequalities described next.

Lemma 3.5 (Strengthened positive-edge inequality). Consider a positive edge $(i, j) \in E^{+}$. Let $S \subseteq O(i)$ be a subset of candidates to represent vertex $i$. The inequality

$$
\begin{equation*}
\sum_{p \in S} x_{i}^{p}+\sum_{p \in O(j) \backslash S} x_{j}^{p} \leq 1, \tag{9}
\end{equation*}
$$

is valid for $P_{G, k}$ and dominates constraints (6) in formulation IP $(G, k)$.
Proof. Consider a vertex $p \in S$ and let $T=O(j) \backslash\{p\}$. Adding inequalities (6) written to each combination of vertices $p$ and $q \in T$, and dividing the result by $|T|$ we obtain the inequality $x_{i}^{p}+(1 /|T|) \sum_{q \in T} x_{j}^{p} \leq 1$. Since $\sum_{q \in T} x_{j}^{p} \in\{0,1\}$, the fractional coefficient in the left-hand side of this inequality can be rounding up giving us $x_{i}^{p}+\sum_{q \in T} x_{j}^{p} \leq 1$. Adding now this inequality written to each $p \in$ $S$, dividing the result by $|S|$ and rounding fractional coefficients in the left-hand side of the obtained inequality, we obtain inequality (9). Clearly constraints (6) are dominated by inequalities (9).

Next, we introduce valid inequalities for $P_{G, k}$ which are related to some substructures of the signed graph. We will see that inequalities (5) and (9) are particular cases of clique-based families of valid inequalities.

### 3.1. Valid inequalities based on subgraph structures

The clique-based inequalities introduced in the next result are a generalization of negative-edge inequalities (5).

Lemma 3.6 (Negative-clique inequality). Let $K^{-} \subseteq V$ be a negative clique in $G$. Consider a vertex $p \in V$ such that $(p, i) \in A$ for each $i \in K^{-}$. The inequality

$$
\begin{equation*}
\sum_{i \in K^{-}} x_{i}^{p} \leq x_{p}^{p} \tag{10}
\end{equation*}
$$

is valid for $P_{G, k}$. This inequality defines a facet for polytope $P_{G, k}$ if and only if $K^{-}$is a maximal negative clique in $G$.

Proof. We use induction on the length of negative clique $K^{-}$. If $\left|K^{-}\right|=1$, the validity follows from constraints (5). Assume that the inequality is valid whenever the negative clique $K^{-}$has a given size $p \geq 1$ and consider a negative clique inequality for which $\left|K^{-}\right|=p+1$. Consider any two vertices $u, v \in$ $K^{-}$. The following inequalities are valid: inequality (10) written to $K^{-} \backslash\{u\}$; inequality (10) written to $K^{-} \backslash\{v\}$; and constraints (5) written to negative edge $(u, v)$. Adding up these three inequalities, dividing the result by two and rounding down the right-hand side of the obtained inequality we conclude the validity proof. Now we prove the facet defining conditions. Consider an arc $(p, i) \in A$ with $i \in K^{-}$. From the incidence vector $x^{I_{p i}}$ and the null vector in $a^{T} x=\alpha$ we conclude that $a_{i}^{p}=-a_{p}^{p}$, for each $i \in K^{-}$. Now consider a vertice $i \in V \backslash\{p\}$. From solution $x^{I_{i}}$ and the null vector we arrive to $a_{i}^{i}=0$. Consider an $\operatorname{arc}(j, i) \in A$ with $i \in K^{-}$and $j \in V \backslash\{p\}$. From the incidence vector $x^{I_{j i}}$ and the null vector we conclude that $a_{j}^{j}+a_{i}^{j}=0$. Since $a_{j}^{j}=0$, we arrive to $a_{i}^{j}=0$. Now let $(i, j) \in A$ be an arc such that $i \in K^{-}$and $j \in V \backslash\{p\}$. From the incidence vector $x^{I_{i j}}$ and the null vector, we conclude that $a_{j}^{i}=0$. Consider an $\operatorname{arc}(p, j) \in A$ with $j \in V \backslash K^{-}$. Since $K^{-}$is a maximal negative clique, there exists a vertice $i \in K^{-}$such that $(j, i) \notin E^{-}$. This allow us to define an incidence vector $x^{I^{\prime}}$ with $I^{\prime}=\{p, j, i\}$. The incidence vectors $x^{I^{\prime}}$ and $x^{I_{p i}}$ in $a^{T} x=\alpha$ lead us to $a_{j}^{p}=0$. Now consider an arc $(j, p)$ with $j \in V \backslash K^{-}$. In that case, from the incidence vector $x^{I_{j p}}$ and the null vector, we conclude that $a_{p}^{j}=0$. Let $(i, j) \in A$ be an arc such that $i, j \in V \backslash K^{-} \cup\{p\}$. From solution $x^{I_{i j}}$ and the null vector, we arrive to $a_{j}^{i}=0$. Finally, assume that there is a vertex $j \in V \backslash K^{-} \cup\{p\}$ such that $(j, i) \in E^{-}$for each $i \in K^{-}$. In this case, the negative-clique inequality defined by $K^{-}$and $p$ is clearly dominated by an inequality in the same family defined by $K^{-} \cup\{j\}$ and $p$.

Positive cliques induce a class of valid inequalities for $P_{G, k}$ that generalizes strongest positive-edge inequalities (9).

Lemma 3.7 (Positive-clique inequality). Let $K^{+} \subseteq V$ be a positive clique in $G$. For each vertex $j \in K^{+}$, consider a set $S(j) \subseteq \bar{O}(j)$. Additionally, assume
that $S(i) \cap S(j)=\emptyset$ for each pair $i, j \in K^{+}$. The inequality

$$
\begin{equation*}
\sum_{j \in K^{+}} \sum_{i \in S(j)} x_{j}^{i} \leq 1, \tag{11}
\end{equation*}
$$

is valid for $P_{G, k}$. This inequality defines a facet for polytope $P_{G, k}$ if the following conditions are satisfied.
(i) $K^{+}$is a maximal positive clique in $G$,
(ii) for each $j \in K^{+}, S(j)$ is a maximal set, i.e., $O(j) \backslash \cup_{p \in K} S(p)=\emptyset$,
(iii) for each $(i, j) \in A$, with $i, j \notin K^{+}$, there is a $p \in K^{+}$such that $(i, p) \notin E^{+}$ and $(j, p) \notin E^{+}$.

Proof. This proof follows the same lines of the proof in Lemma 3.6. For the validity proof, we use induction on the length of positive clique $K^{+}$and the validity of constraints (9). For the facet-defining proof, as in the proof of the Lemma 10, we use the incidence vectors of partitions (i.e. feasible solutions belonging to the facet) enumerated in Table 1. Remember the facet-defining proof methodology stated in the beginning of this section. Each double line of Table 1 is associated with a set of components of the coefficient vector $a \in \Re^{n+m=\left|A \cup A^{0}\right|}$. The first column indicates the set of components (or arcs) being considered. The second column exhibits two sets defining incidence vectors satisfying inequality (11) with equality, while the third column states the conclusion obtained. The fourth column presents remarks linking the construction of the incidence vectors used in this proof with facet-defing conditions (i)-(iii). In Table 1, $S\left(K^{+}\right)=\cup_{p \in K} S(p)=\emptyset$ and, for each $i \in S\left(K^{+}\right)$, we define $p_{i}=j \in K^{+}$such that $i \in S(j)$.

The next valid inequality is a generalization of inequalities (11).
Lemma 3.8 (Negative-positive-clique inequality). Let $K^{-} \subseteq V$ and $K^{+} \subseteq V$ be, respectively, a negative and a positive clique in $G$ such that, for each pair of vertices $i \in K^{-}$and $j \in K^{+},(i, j) \in E^{+}$. Let $p \in V$ be a vertex such that, for each $i \in K^{-},(p, i) \in A$. Moreover, for each vertex $j \in K^{+}$, consider a set $S(j) \subseteq O(j) \backslash\{p\}$. Additionally, assume that $S(i) \cap S(j)=\emptyset$ for each pair $i, j \in K^{+}$. The inequality

$$
\begin{equation*}
\sum_{i \in K^{-}} x_{i}^{p}+\sum_{j \in K^{+}} \sum_{i \in S(j)} x_{j}^{i} \leq 1 \tag{12}
\end{equation*}
$$

is valid for $P_{G, k}$.
Proof. We use induction on the length of negative clique $K^{-}$. If $\left|K^{-}\right|=1$, the inequality is a positive-clique inequality and the validity follows from Lemma 3.7. Assume that the inequality is valid whenever the negative clique $K^{-}$has a given size $p \geq 1$. Now, consider a negative-positive-clique inequality for which

| $(i, j) \in A \cup A^{0}$ | Partitions | Conclusion | Remark |
| :--- | :--- | :--- | :--- |
| $(i, i), i \in K^{+}$ | $I_{i}$ | $a_{i}^{2}=\alpha$ |  |
|  | $I_{i^{\prime}}, i^{\prime} \in K^{+}$ |  |  |
| $(i, i), i \in S\left(K^{+}\right)$ | $I_{p}$ | $a_{i}^{i}=0$ | $\exists p \in K^{+}$s.t. (i,p) $\notin E^{+}$due to (i) |
|  | $\{\{i\},\{p\}\}$ |  |  |
| $(i, i), i \in V \backslash\left(K^{+} \cup S\left(K^{+}\right)\right)$ | $I_{p}$ | $a_{i}^{i}=0$ | $\exists p \in K^{+}$s.t. (i,p) $\notin E^{+}$due to (i) |
|  | $\{\{i\},\{p\}\}$ |  |  |
| $(i, j), j \in K^{+}, i \in S(j)$ | $I_{i j}, I_{j}$ | $a_{j}^{i}=\alpha$ |  |
| $(i, j), j \in K^{+}, i \in S\left(K^{+} \backslash j\right)$ | $\left\{\left\{i, j, p_{i}\right\}\right\}$ | $a_{j}^{i}=0$ |  |
| $(i, j), j \in K^{+}, i \in V \backslash K^{+} \backslash S\left(K^{+}\right)$ |  | $I_{i p_{i}}$ |  |
| $(i, j), j \in S\left(K^{+}\right), i \in K^{+}$ | $I_{i j}, I_{i}$ | $a_{j}^{i}=0$ |  |
| $(i, j), j \in S\left(K^{+}\right), i \in S\left(K^{+}\right)$ | $\{\{i, j\},\{p\}\}$ | $a_{j}^{i}=0$ | $p$ exists due to (iii) |
|  | $I_{p}$ |  |  |
| $(i, j), j \in S\left(K^{+}\right), i \in V \backslash K^{+} \backslash S\left(K^{+}\right)$ | $\{\{i, j\},\{p\}\}$ | $a_{j}^{i}=0$ | $p$ exists due to (iii) |
|  | $I_{p}$ | $I_{i j}, I_{i}$ | $a_{j}^{i}=0$ |
| $(i, j), i \in K^{+}, j \in V \backslash K^{+} \backslash S\left(K^{+}\right)$ | $\{\{i, j\},\{p\}\}$ | $a_{j}^{i}=0$ | $p$ exists due to (iii) |
| $(i, j), i \in S\left(K^{+}\right), j \in V \backslash K^{+} \backslash S\left(K^{+}\right)$ | $I_{p}$ |  |  |
| $(i, j), i, j \in V \backslash K^{+} \backslash S\left(K^{+}\right)$ | $\{\{i, j\},\{p\}\}$ | $a_{j}^{i}=0$ | $p$ exists due to (iii) |

Table 1: Sketch of feasible solutions used in the facet-defining proof of Lemma 3.7.
$\left|K^{-}\right|=p+1$. Consider a vertex $u \in K^{-}$. The following inequalities are valid: $\sum_{i \in K^{-} \backslash\{u\}} x_{i}^{p}+\sum_{j \in K^{+}} \sum_{i \in S(j)} x_{j}^{i} \leq 1$ from the inductive hypothesis; $x_{u}^{p}+$ $\sum_{j \in K^{+}} \sum_{i \in S(j)} x_{j}^{i} \leq 1$ from Lemma 3.7; and $\sum_{i \in K^{-}} x_{i}^{p} \leq 1$ from Lemma 3.6. We finish this proof by adding up these inequalities, dividing the result by two and rounding down the right-hand side of the obtained inequality.

Holes in $G$ can also give rise to a family of valid inequalities for $P_{G, k}$.
Lemma 3.9 (Hole inequality 1). Let $H \subseteq V$ be a hole in $G$ with $t$ positive edges. Assume that $|H|+t$ is odd. Let $p \in V$ be a vertex such that, for each $i \in H,(p, i) \in A$. The inequality

$$
\begin{equation*}
\sum_{i \in H} x_{i}^{p} \leq\lfloor(|H|+t) / 2\rfloor x_{p}^{p}, \tag{13}
\end{equation*}
$$

is valid for $P_{G, k}$.
Proof. Let $\bar{x}$ be a feasible solution in $P_{G, k}$. If $\bar{x}_{p}^{p}=0$, the validity follows from inequalities (4) written for each $\operatorname{arc}(p, i) \in A$ such that $i \in H$. Assume now that $\bar{x}_{p}^{p}=1$. Add up inequalities (5) written for each $\operatorname{arc}(i, j) \in E^{-}$such that $i, j \in H$, and the inequalities $x_{i}^{p}+x_{j}^{p} \leq 2 x_{p}^{p}$ written for each arc $(i, j) \in E^{+}$ such that $i, j \in H$. The validity follows from dividing the obtained inequality by two and rounding down its right-hand side.

Lemma 3.10 (Hole inequality 2). Let $H \subseteq V$ be a hole in $G$ with $t$ negative edges. Assume that $|H|+t$ is odd. For each vertex $j \in H$, consider a set
$S(j) \subseteq O(j)$. Additionally, assume that $S(i) \cap S(j)=\emptyset$ for each pair $i, j \in H$ such that $(i, j) \in E$. The inequality

$$
\begin{equation*}
\sum_{j \in H} \sum_{i \in S(j)} x_{j}^{i} \leq\lfloor(|H|+t) / 2\rfloor, \tag{14}
\end{equation*}
$$

is valid for $P_{G, k}$.
Proof. This validity proof follows exactly the lines of the validity proof in Lemma 3.9: add up the valid inequalities $\sum_{p \in S(i)} x_{i}^{p}+\sum_{p \in S(j)} x_{j}^{p} \leq 1$ written for each arc $(i, j) \in E^{+}$such that $i, j \in H$, and the inequalities $\sum_{p \in S(i)} x_{i}^{p}+$ $\sum_{p \in S(j)} x_{j}^{p} \leq 2$ written for each arc $(i, j) \in E^{-}$such that $i, j \in H$. The validity follows from dividing the obtained inequality by two and rounding down its right-hand side.

A signed graph is clusterable if and only if it does not contain a cycle with exactly one negative edge [17]. This property motivates the following family of valid inequalities.

Lemma 3.11 (1-edge negative cycle). Let $C \subseteq V$ be a cycle in $G$ with exactly one negative edge. For each $j \in C$, let us denote by $j^{-}, j^{+} \in C$ the two vertices in $C$ adjacent to $j$. Let us define, for each $j \in C, a_{j}^{j}=2$ if (i) $\left(j^{-}, j\right),\left(j, j^{+}\right) \in$ $E^{+}$and (ii) $j>j^{-}, j^{+} ; a_{j}^{j}=1$ otherwise. The inequality

$$
\begin{equation*}
\sum_{j \in C}\left(a_{j}^{j} x_{j}^{j}+\sum_{i \in O(j) \backslash\{j\}} x_{j}^{i}\right) \leq|C|-1 \tag{15}
\end{equation*}
$$

is valid for $P_{G, k}$.
Proof. By definition, a $k$-balanced signed graph is clusterable. Hence, the following inequality is clearly valid:

$$
\begin{equation*}
\sum_{j \in C}\left(x_{j}^{j}+\sum_{i \in O(j) \backslash\{j\}} x_{j}^{i}\right) \leq|C|-1 . \tag{16}
\end{equation*}
$$

Let $p \in C$ be a vertex satisfying (i) and (ii) (see Figure 2). Next, we argue that inequality (16) can be strengthened by making $\alpha_{p}^{p}=2$ in the the following inequality:

$$
\begin{equation*}
\alpha_{p}^{p} x_{p}^{p}+\sum_{i \in O(p) \backslash\{p\}} x_{p}^{i}+\sum_{j \in C \backslash\{p\}}\left(x_{j}^{j}+\sum_{i \in O(j) \backslash\{j\}} x_{j}^{i}\right) \leq|C|-1 . \tag{17}
\end{equation*}
$$

The assignment $x_{p}^{p}=0$ do not impose a limit to the value of $\alpha_{p}^{p}$. However,


Figure 2: A cycle $C$ with exactly one negative edge: at most $|C|-1$ vertices can be simultaneously in a feasible solution. If vertex $p$ is a representative one, vertices $p^{-}$and $p^{+}$are prevented to be in the solution.
whenever $x_{p}^{p}=1$,

$$
\alpha_{p}^{p} \leq(|C|-1)-\sum_{i \in O(p) \backslash\{p\}} x_{p}^{i}-\sum_{j \in C \backslash\{p\}}\left(x_{j}^{j}+\sum_{i \in O(j) \backslash\{j\}} x_{j}^{i}\right) .
$$

If $p$ is a representative vertex $\left(x_{p}^{p}=1\right)$ and (i) and (ii) are satisfied, vertices $p^{-}, p^{+}$are forbidden from belonging to a solution. Hence, the highest value the right-hand side of the previous inequality can assume is 2 and happens whenever all $|C|-3$ vertices in $C \backslash\left\{p, p^{-}, p^{+}\right\}$belong to the solution. Now, if we consider the valid inequality (17) with $\alpha_{p}^{p}=2$, the same procedure can be applied iteratively on each vertex $q \in C$ satisfying (i) and (ii) (one must observe that the coefficients equal to 2 in the inequality being strengthened do not invalidate the last conclusion about its right-hand side).

A star in $G$ is a tree $X_{l}=\left(V^{X}, E^{X}\right)$ with one internal node and $l$ leaves. A positive star is a star such that $E^{X} \subseteq E^{+}$. Positive stars also give rise to a family of valid inequalities for $P_{G, k}$.
Lemma 3.12 (Positive-star inequality). Let $X_{l}=\left(V^{X}, E^{X}\right)$ be a positive star in $G$ with internal node $p$ such that $k \geq l$ and, for each $j \in V^{X} \backslash\{p\}, p>j$. For each vertex $j \in V^{X} \backslash\{p\}$, consider a set $S(j) \subseteq O(j)$. Additionally, assume that $S(i) \cap S(j)=\emptyset$ for each pair $i, j \in V^{X} \backslash\{p\}$. The inequality

$$
\begin{equation*}
\sum_{j \in V^{X} \backslash\{p\}}\left((k-1) x_{p}^{j}+\sum_{i \in S(j)} x_{j}^{i}\right)+k x_{p}^{p} \leq k, \tag{18}
\end{equation*}
$$

is valid for $P_{G, k}$.


Figure 3: A positive star $X_{3}=\left(V^{X}, E^{X}\right)$ with internal node $p$ such that $k \geq 3$. The leafs of the positive star cannot belong to the solution whenever $p$ is a representative vertex. On the other hand, $j^{\prime}$ can represent $p$ and in that case any other leaf of the positive star is prevented to be in the feasible solution.

Proof. Due to constraints (2), $\sum_{j \in V^{x} \backslash\{p\}} x_{p}^{j}+x_{p}^{p} \leq 1$, which implies

$$
\begin{equation*}
\sum_{j \in V^{X} \backslash\{p\}}(k-1) x_{p}^{j}+k x_{p}^{p} \leq k . \tag{19}
\end{equation*}
$$

Due to constraints (2) and since $l \leq k$ we have also

$$
\begin{equation*}
\sum_{j \in V^{X} \backslash\{p\}} \sum_{i \in S(j)} x_{j}^{i} \leq k . \tag{20}
\end{equation*}
$$

Since $X_{l}$ is a positive star, the left-hand side of (19) equal to $k\left(x_{p}^{p}=1\right)$ implies the left-hand side of (20) is equal to zero. Likewise, the left-hand side of (19) equal to $k-1$ (See Figure $3 ; x_{p}^{j^{\prime}}=1, j^{\prime} \in V^{X} \backslash\{p\}$ ) implies the left-hand side of (20) less than or equal to one (being one whenever $j^{\prime} \in S\left(j^{\prime}\right)$ ).

## 4. A Branch-and-cut approach

The branch-and-cut algorithm implemented in this work is composed by four basic components: the initial formulation, the primal heuristic, the cut generation and the branching rule. Next, we describe the main ingredients of this approach, used in the computational experiments described in Section 5. The strategies described here were defined after a number of computational experiments have been performed.

The initial formulation is obtained from the formulation presented in Section 2 by replacing constraints (6) for strengthened ones: from Lemma 3.5, inequalities (6) can be replaced by inequalities (9).

### 4.1. Primal heuristic: An ILS metaheuristic

Iterated Local Search (ILS) [42] is a metaheuristic that aims to improve upon stochastic multi-start search by sampling in a distant neighborhood of candidate solutions, and then applying a local search technique to refine solutions to their local optima. To reach these distant neighborhoods, it explores a sequence of solutions created as perturbations of the current best solution, the result of which is refined using an embedded heuristic. In this context, we have designed and implemented an ILS similar to [24], that executes in sequence three phases: (i) a random/greedy heuristic in a constructive phase, (ii) a Variable Neighborhood Descent (VND) [44] in the local search phase and (iii) perturbation moves as a diversification mechanism. After each successful perturbation move, the ILS metaheuristic returns to phase (ii). In the experiments, the ILS executes 5 multi-start iterations (i.e., the number of times the sequence is re-started), with a stopping criteria of 20 perturbations (i.e., number of perturbations applied without improvement). In the remainder of this section, we describe these three phases.

The input for the metaheuristic is the signed graph $G=(V, E, s)$. We assume that a feasible solution for the problem is defined by a family of $k$ disjoint sets $\left(V_{1}, \ldots, V_{k}\right)$ such that $V_{1}, \ldots, V_{k} \subseteq V, \bigcup_{i=1, \ldots, k} E\left[V_{i}\right] \subseteq E^{+}$and $\left(E\left[V_{1} \cup \ldots \cup V_{k}\right]\right) \backslash\left(\bigcup_{i=1, \ldots, k} E\left[V_{i}\right]\right) \subseteq E^{-}$.

Construction phase. The overall method attempts to find a maximal feasible solution $\left(V_{1}, \ldots, V_{k}\right)$. The construction phase begins with $V_{i}=\emptyset$, for each $i=1, \ldots, k$, and build these sets in a random/greedy way by adding one new vertex to the solution at a time. Let $\left(V_{1}, \ldots, V_{k}\right)$ be a feasible solution. We define $\operatorname{Cand}\left(V_{i}\right)=\left\{u \in V \backslash \cup_{i=1, \ldots, k} V_{i} \mid\left(V_{1}, \ldots, V_{i} \cup\{u\}, \ldots, V_{k}\right)\right.$ is a feasible solution\}, the set of vertices that are candidate to be added to set $V_{i}$. At each iteration, a vertex $u \in \operatorname{Cand}\left(V_{i}\right)$ (for $i=1, \ldots, k$ ) is randomly selected and a new set $V_{i}^{\prime}$ is reached by inserting vertex $u$ into the set $V_{i}$. The procedure is repeated until $\operatorname{Cand}\left(V_{i}\right)=\emptyset$, for all $i=1, \ldots, k$.

Local search phase. Let $N_{\propto}$ be an ordered set of predefined neighborhood structures. Whenever one neighborhood fails to improve the current solution, the VND picks the next neighborhood in $N_{\propto}$ to continue the search. The local search phase halts when no better solution is found in the set of neighborhood structures of the current solution. The ILS is composed by the following 5 ordered neighborhoods:

Hire: inserting a vertex $v \in\left(\operatorname{Cand}\left(V_{1}\right) \cup \ldots \cup \operatorname{Cand}\left(V_{k}\right)\right)$ into the corresponding subset (i.e., $V_{1}$ if $v \in \operatorname{Cand}\left(V_{1}\right), V_{2}$ if $v \in \operatorname{Cand}\left(V_{2}\right), \ldots$ ). The process is repeated until $\left(\operatorname{Cand}\left(V_{1}\right) \cup \ldots \cup \operatorname{Cand}\left(V_{k}\right)\right)=\emptyset$.
$\operatorname{Drop}(\mathbf{i})$-Hire: removing a vertex $u$ from $V_{i}$ (i.e., $V_{i}=V_{i} \backslash\{u\}$ ) and Hire.
Move(i,j)-Hire: moving a vertex $u$ from $V_{i}$ to $V_{j}$ (i.e., $V_{i}=V_{i} \backslash\{u\}$ and $\left.V_{j}=V_{j} \cup\{u\}\right)$ and Hire.

Switch(i,j)-Hire: switching vertices $u$ from $V_{i}$ and $v$ from $V_{j}$ (i.e., $V_{i}=$ $\left(V_{i} \backslash\{u\}\right) \cup\{v\}$ and $\left.V_{j}=\left(V_{j} \backslash\{v\}\right) \cup\{u\}\right)$ and Hire.

2Drop(i)-Hire: removing two vertex $u_{1}$ and $u_{2}$ from $V_{i}$ (i.e., $V_{i}=$ $\left.V_{i} \backslash\left\{u_{1}, u_{2}\right\}\right)$ and Hire.

It is important to emphasize that only feasible movements are accomplished. Notice that the neighborhood Hire applied on a solution created by the constructive phase is empty.

Perturbation moves. We perform multiple Drop, Move and Switch movements randomly chosen in such a way that the resulting modification is sufficient to escape from local optima and analyze different regions of the search space.

The multistart ILS is executed before the beginning of the branch-and-cut method: the solution obtained is used as a first primal bound. Additionally, at each integer solution found in the branch-and-cut tree, phases (ii) and (iii) are repeated by using this integer solution as the start solution (with the same stopping criteria described here).

### 4.2. Cut generation

After a linear programming problem has been solved in the branch-and-cut tree, the algorithm checks if the solution is integer feasible. If this is not the case, the cut generation procedure is called and a set of separation routines is executed. A limit of 300 cuts per iteration is set (with a limite of 200 per type of cut).

The cut generation procedure is based on the inequalities described in Section 3, whose separation consists basically of finding cliques, holes, and stars in the graph which maximize a given measure. It follows the scheme and procedures suggested in the branch-and-cut algorithm proposed in [28] for solving the partition coloring problem.

The separation problem associated with cuts based on clique structures (9), (10), (11) and (12) is an NP-Hard problem (maximum clique problem [29]). We use GRASP [46] heuristics for finding cuts based on maximal clique structures. The GRASP procedure is an iterative method composed of two phases: a construction phase and a local search phase. The construction phase finds an initial solution that may be later improved by the local search phase. Roughly speaking, this heuristic iteratively build an initial solution (clique) in a greedy random way. At each step of the construction phase, one vertex is inserted into the clique with a probability equivalent to its contribution to the cut violation, until a maximal clique is found. Later, the initial solution may be improved by a local search procedure, where the neighborhood is defined as the set of all cliques obtained by exchanging only one vertex in the current solution. Again, the criterion for the acceptance of an improved solution is the contribution to the violation of the cut being separated. For inequalities (11), sets $S(j), j \in K^{+}$, are generated in a way that condition (ii) of Lemma 3.7 is satisfied.

Considering a value $z_{i}$ associated to each $i \in V$, odd-hole inequalities

$$
\sum_{i \in H} z_{i} \leq\lfloor(|H|+t) / 2\rfloor
$$

can be separated in polynomial time, as described in [35], by solving a set of $n$ shortest path problems in a bipartite graph with vertex sets $V$ and $V^{\prime}$ (see Section 6.5 from [35] for the graph construction). Following this procedure, inequalities (13) can be separated in polynomial time by: defining $z_{i}=x_{i}^{p} / x_{p}^{p}$, whenever $x_{p}^{p}>0$; redefining the arc set of the bipartite graph (positive edges are associated with arcs linking vertices in a same set $V$ or $V^{\prime}$; negative edges with arcs linking vertices in different sets); and adding 1 to the cost of each arc in the bipartite graph associated with a positive edge in the original graph. In the same way, inequalities (14) can also be separated in polynomial time whenever sets $S(j)$ are fixed. As it is explained in [35], this is a textbook solution that provides at most one odd hole after the solution of each shortest path problem. Then, the Hoffman and Padberg heuristic, described in [35], is used here for finding odd holes (13) and (14). Basically, the method performs a breadthfirst search labeling of the vertices, starting from any root vertex $r$ randomly chosen. A label $h_{v}$ is assigned to each vertex $v \in V \backslash\{r\}$ representing the edge distance between $r$ and $v$. For any two adjacent vertices $v_{1}$ and $v_{2}$ with labels $h_{v_{1}}=h_{v_{2}} \geq 2$, if there exist two vertex-disjoint shortest paths (from $v_{1}$ to $r$ and from $v_{2}$ to $r$ ), then there exists an odd cycle that contains vertices $v_{1}, v_{2}$, and $r$, which can be inspected in order to find violated cuts.

Let us define $z_{i}=a_{j}^{j} x_{j}^{j}+\sum_{i \in O(j) \backslash\{j\}} x_{j}^{i}$. The 1-edge negative cycle inequality (15) can be re-written as $\sum_{i \in C} z_{i} \leq|C|-1$, where $C$ is a cycle in $G$ with exact one negative edge. The separation problem associated with this inequality is polynomially solved by finding $n$ shortest paths in an auxiliar directed graph $D^{a}=\left(V_{1}^{a} \cup V_{2}^{a}, A^{a}\right)$ defined next. For each $i \in V$, we create a vertice $i \in V_{1}^{a}$ and a vertice $i^{\prime} \in V_{2}^{a}$. For each positive edge $(i, j) \in E^{+}$, we create four arcs $(i, j),(j, i),\left(i^{\prime}, j^{\prime}\right),\left(j^{\prime}, i^{\prime}\right) \in A^{a}$ while for each negative edge $(i, j) \in E^{-}$, we create two $\operatorname{arcs}\left(i, j^{\prime}\right),\left(j, i^{\prime}\right) \in A^{a}$. A length $d_{i j}=1-z_{i}$ is defined for each $\operatorname{arc}(i, j) \in A^{a}$. The separation problem of 1-edge negative cycle inequalities is solved by looking for each minimum path from $i \in V_{1}^{a}$ to $i^{6} \in V_{2}^{a}$ in $D^{a}$. Notice that, each path from $i \in V_{1}^{a}$ to $i^{6} \in V_{2}^{a}$ defines a cycle in $G$ with exactly one negative edge. If such a path has a length smaller than one, a violated inequality (15) has been found, otherwise each such inequality is satisfied. Again, this is a textbook solution and thus, the separation of inequalities (15) is incorporated in the Hoffman and Padberg heuristic described above.

The separation problem of the positive-star inequalities (18) is NP-Hard. Suppose that we have a fixed positive star $X_{l}=\left(V^{X}, E^{X}\right)$ with internal vertex $p$. Even the problem of deciding if there exist, for each leaf $j \in V^{X} \backslash\{p\}$, a set $S(j) \subseteq O(j)$ that violates constraint (18) is an NP-complete problem: it consists in solving the maximum stable set problem [29] defined on graph $H=\left(V^{H}, E^{H}\right)$ described next. For each $i \in S(j)$ and for each $j \in V^{X} \backslash\{p\}$, create a vertex $i j$ in $V^{H}$. For each pair of vertices $i j_{1}, i j_{2} \in V^{H}$ create an edge $\left(i j_{1}, i j_{2}\right) \in$
$E^{H}$. Therefore, our separation routine for inequalities (18) consists in partially enumerating the positive stars and in using the same GRASP approach from clique cuts to find stable sets in graph $H=\left(V^{H}, E^{H}\right)$. Finally, if no violated inequality is found by all these separation routines, we construct a conflict graph [5] representing all the conflicts among variables $x_{i j},(i, j) \in A^{0}$, and run a GRASP procedure looking for cliques in the conflict graph.

At the end of the cut generation phase, if either no violated inequality is found or a limit of $(|V| / 10)$ cut generations rounds is reached without an improvement of at least 0.01 in the cost function, the algorithm enters in the branching phase described next.

### 4.3. Branching rule

The standard $0-1$ branching rule can be very asymmetrical when representatives formulation is used, producing unbalanced enumerative trees. Our branching rule is based on constraints (2). The intuition behind this constraint based branching is the generation of more balanced enumerative trees. In the following, let $\bar{x} \in \Re^{n+m}$ be a fractional solution to the $k$-MBS problem obtained at a given node. Let $p \in V$ be a vertex such that $\bar{x}_{p}^{j}$ is fractional for some $j \in V \backslash p$. First, we define a partition $S_{1}, S_{2}$ for the set $O(p)$ such that $\sum_{j \in S 1} x_{p}^{j}$ (and $\sum_{j \in S 1} \bar{x}_{p}^{j}$ ) is a fractional value. Sets $S_{1}$ and $S_{2}$ are defined in a way that $\left|\sum_{j \in S 1} x_{p}^{j}-\sum_{j \in S 2} x_{p}^{j}\right|$ is minimized. We create three branches on the branching tree by adding constraints defined on sets $S_{1}$ and $S_{2}$ :
$\sum_{j \in S 1} x_{p}^{j}=1, p$ is in the solution and is represented by a vertex in $S_{1}$,
$\sum_{j \in S 2} x_{p}^{j}=1, p$ is in the solution and is represented by a vertex in $S_{2}$,
$\sum_{j \in O(p)} x_{p}^{j}=0, p$ does not belong to the solution.
The branching strategy adopted in our branch-and-cut approach is the following. The standard $0-1$ branching rule is used whenever there is a $p \in V$ such that $\bar{x}_{p}^{p}$ is fractional; otherwise the constraint based branching is applied. The branch-and-cut tree is investigated with the best-bound-first strategy.

## 5. Computational experiments

In this section, we report the computational experiments carried out with the formulation and methods described in the last sections.

### 5.1. Test problems

In order to test the performance of our branch-and-cut approach, we considered two sets of instances:

- a set of 126 signed networks extracted with the algorithm described in [24] from the UNGA voting records of the separate annual sessions (from 1946 to 2008);
- a set of 180 random instances with a predefined community structure.

Random instances. We generated 180 random signed networks with a predefined community structure inspired in the algorithm used in [30]. A random signed network is defined as $S G\left(c, n, p_{-}, p_{+}, \epsilon\right)$ where parameters $c, n, p_{-}$and $p_{+}$are used to generate a balanced signed graph: $c$ is the number of communities in the network, $n$ is the number of nodes equally distributed in $c$ communities, $p_{-}$denotes the probability of negative links appearing between communities, and $p_{+}$denotes that of positive links appearing within communities. Based only on these parameters, the balanced signed graph generated presents around $c \times \frac{n}{2 c}\left(\frac{n}{c}-1\right) \times p_{+}$positive links internal to the $c$ communities and $\frac{c(c-1)}{2} \times$ $\frac{n}{2 c} \frac{n}{c} \times p_{-}$negative links among the $c$ communities, with each vertice having around $\left(\frac{n}{c}-1\right) \times p_{+}+\left(\frac{n}{c}-1\right) \times p_{+}$neighbors. The last parameter is used to introduce a deviation from the balanced state: each generated signed link goes through a sign shift with a probability $\epsilon$. The parameters $c, n$ and $\epsilon$ take values in the following sets: $c \in\{2,3,4,5\}, n \in\{60,70,80\}$ and $\epsilon \in\{0.5,0.1,0.2\}$. The parameters $p_{-}$and $p_{+}$are fixed, respectively, as 0.3 and 0.05 giving us signed graphs such that each vertice has between 6 and 11 neighbors. A number of five instances is generated for each combination of these parameters.

UNGA instances. The class UNGA is composed of instances derived from the community structure of networks representing voting on resolutions in the United Nations General Assembly, for 63 different sessions. The algorithm proposed in [24] uses two parameters, $\alpha$ and $\beta$, to control the decision about the addition of negative and positive edges in the network representing a vote session. Two sets of 63 instances were extracted by setting $\alpha=0.3, \beta=0.5$ for a first set (Set 3.5), and $\alpha=0.1, \beta=0.8$ for a second set (Set 1.8). See [24] for details about the extraction of these signed networks. Contrary to our supposition in Section 1, UNGA signed networks have a few parallel edges. However, all the methods and results developed in this paper can be applied to the solution of the $k$-mbs problem on networks with parallel edges.

Table 4 describes the main characteristics of both sets of UNGA instances. Columns of this table give us the following information: the instance name indexed by the number of the vote session, the number of vertices " $n$ ", and, according to the set of instances, the number of positive edges " $m^{+}$", the positive density $d^{+}=2 m^{+} / n(n-1)$ of the network, the number of negative edges " $m^{-}$" and the negative density $d^{-}=2 m^{-} / n(n-1)$ of the network. Given a vote session, the signed network extracted by setting $\alpha=0.3, \beta=0.5$ is a subgraph of the network extracted by setting $\alpha=0.1, \beta=0.8$. According to Table 4, a network in set 1.8 has, in average, $23 \%$ more positive edges and $80 \%$ more negative edges, than the equivalent network in set 3.5.

### 5.2. Methods and computational environment

We implemented and compared three exact methods based on the ILP formulation introduced in this work:

- BC: Branch-and-cut algorithm as described in Section 4.
- $\mathrm{BB}+01$ : Branch-and-bound algorithm with formulation described in Section 2, standard $0-1$ branching rule and primal bounds given by the ILS metaheuristic described in Section 4.1.
- BB+S1S2: Branch-and-bound algorithm with the formulation described in Section 2, constraint branching rule described in Section 4.3 and primal bounds given by the ILS metaheuristic described in Section 4.1.

All these methods were coded in ANSI C ++ by using Xpress-MP Optimizer libraries version 27.01 running on a Intel, 3.6 GHz , equipped with 32 GByte RAM and a Linux operating system. In order to obtain the primal bound $\alpha\left(G^{-}\right)$established in Proposition (2.1), we have used the same GRASP approach implemented for the separation of positive-star inequalities. The CPU time limit is set to 1 h for each solution method.

For the sake of space, we do not exhibit in this text all the results obtained in our experiments. Nevertheless, all the instances used and all the results (including additional plots and tables) mentioned here are available in http: //www.ic.uff.br/~yuri/files/kmbs.zip.

### 5.3. Numerical Results

Next, we present and discuss the results obtained on Random instances (Section 5.3.1) and on UNGA instances (Section 5.3.2). Besides evaluating the branch-and-cut algorithm described in Section 4, we also investigate the efficiency of the constraint branching rule described in Section 4.3 for the representatives formulation. For that purpose, we try to solve the instances in our test sets by using algorithms $\mathrm{BB}+01, \mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ and the branch-and-cut algorithm BC for $k \in\{2,3,4,|V|\}$.

### 5.3.1. Random instances

$B B$ methods vs $B C$. Figures 4 and 5 present plots comparing the performance obtained by the three exact approaches on the set of 180 random instances, for different values of $k$ and $\varepsilon$. In these profile plots, the performance of each method is described by a curve. For a given value in axis $x$, the corresponding value in axis $y$ informs the proportion of instances for which the time spent by the exact method was not $x$-times worst than the time spent by the faster approach. From 4, we can see that the performance of $\mathrm{BB}+01$ method remains the same as $k$ increases while the performance of $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ method deteriorates, practically equating to $\mathrm{BB}+01$ method when $k=n$. From 5 , we arrive to the same conclusion as $\varepsilon$ increases. Clearly, the BC method obtained the best performance among the three methods. We can also observe that, the performance of $\mathrm{BB}+01$ and $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ are closer to the performance of BC method as parameter $\varepsilon$ increases, i.e., as the deviation from a balance state increases.

Tables 2 and 3 summarize the results obtained with the three exact approaches on the set of random instances. These tables display, respectively, the total number of instances solved to optimality and the average gaps on unsolved
instances (i.e. instances not solved to optimality in the time limit). The first multicolumn on these tables gives the value of $k$ and the method executed; the next three multicolumns exhibit the information on instances grouped according to the value of $n, C$ and $\epsilon$. From these tables, we conclude that $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ has slightly better performance than $\mathrm{BB}+01$ with respect to the 3 criteria used in the comparison. However, BC has much better performance than both versions of BB (i.e. $\mathrm{BB}+01$ and $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ ) with respect to the 3 criteria used in the comparison: more instances solved in less time and presenting small gaps for unsolved instances. BC was able to solve $\{115,97,109,100\}$ instances when $k \in\{2,3,4, n\}$ against only $\{19,1,0,2\}$ and $\{70,18,9,3\}$ from, respectively, $\mathrm{BB}+01$ and $\mathrm{BB}+$ S1S2. Moreover, the performance of BC with respect to the number of instances solved to optimality as well as the time spent with these instances is less sensitive to parameters $n, \epsilon$ and $k$, than $\mathrm{BB}+01$ and $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$.
$B C$ method. Figures 6 and 7 explore the results obtained by BC. The graphics in Figure 6 display, for $k=2$ and for $k=3$ : the values of the linear relaxations of the first model ("Init Form LR") and at the root of the search tree ("Root LR"), the value of the upper ("UB") and lower bound ("LB") at the end of the execution, and also the value of the lower bound provided by Proposition 2.1 ("Stable(G-)"). The quality of the results obtained for $k \in\{4,|V|\}$ are very similar to the ones obtained for $k=3$. We can verify that the linear relaxation of the representatives formulation is very poor and deteriorates for $k \geq 3$. Most part of the improvement in the upper bound (due to the inclusion of valid and facet defining inequalities) is achieved in the root of the search tree.

The box-plot graphics in Figure 7 display, for each value of $k$, the percentage of time spent by the cut generation phase ("\%TimeSep"), the percentage of cuts based on maximal clique structures ("\%Cliques), on odd holes and odd anti-holes ("\%Hole"), on cycles ("\%Cycle"), on stars ("\%Star") and on clique inequalities found by the conflict graph separation procedure ("\%Conflict"). We can see that the time spent by the separation procedure is never more than $21 \%$ of the execution time. Most part of the cuts generated are clique inequalities with percentage values usually superior than $90 \%$ with almost all clique cuts generated by the GRASP procedure; the additional conflict graph separation procedure is almost never needed. Motivated by these results, we run the BC method only using the GRASP procedure for separation of clique inequalities in the cut generation phase. These additional results ${ }^{2}$ show the performance of the BC deteriorated drastically: around less $60 \%$ of instances solved to optimality with worst average gaps for unsolved instances. We also run additional experiments in order to study the relevance of star and clique inequalities, the less found in the separation phase. These additional results ${ }^{2}$ show star inequalities (only 1616 cuts on $59 / 720$ problems solved) can make a difference in the time spent to solve some instances but, in average, they do not change neither the gap for unsolved instances nor the number of instances solved. When conflict inequalities are not used, the number of instances not solved to optimality increases. Also, the average gaps and the average dual bounds on the root node slightly deteriorates for $k=n$.

ILS metaheuristic. Even if the goal of this work is not in the development of efficient heuristic approaches, one could ask about the performance of the ILS metaheuristic implemented. Four plots were included in the additional material $^{2}$, comparing ILS results on the set of random instances with lower bounds provide by the BC method. We can conclude that the ILS metaheuristic performs well matching, for many instances, the same primal bound of BC. However, this performance deteriorates as the value of $k$ increases and as the number of vertices decreases: for $k=n$ and $n=60$ they are always different except for $5 / 60$ instances.

### 5.3.2. UNGA instances

$B B+01$ vs $B B+S 1 S 2$. Table 5 displays results obtained with $\mathrm{BB}+01$ and $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ on UNGA instances. The first multicolumn on this table provides the value of $k$ and the method executed. The next multicolumns exhibit the results obtained on instances 1.8 and 3.5: "Primal ${ }_{\text {best }}$ " presents the average value of the best integer solution found in the search tree; "Gap \%" presents the average final percentage gap (between the best solution found and the upper bound) whenever the time limit is reached, "Time" presents the average time (in seconds) spent with the instances solved to optimality, "Nodes" presents the average number of nodes explored in the branch-and-bound tree, while "\#Solved" presents the total number of instances solved to optimality by each method. We can see that the new branching rule allows us to solve more instances to optimality and, on average, is less time consuming and explores less nodes in the branch-and-bound tree. The quality of the average gap has been maintained when $k \in\{2,3,4\}$ and has been deteriorated when $k=n$. However, as we will see in the following, this deterioration is explained by the fact that the primal heuristic used does not provide good lower bounds when $k=n$. Moreover, when $k=n$, the branch-and-bound methods halt with out of memory error for five instances (S57 to S62).
$B C$ method. Now, we discuss the performance of the branch-and-cut algorithm BC over the test sets. Tables 6 and 7, report, respectively, the results obtained with instances in sets 3.5 and 1.8. In Tables 6 and 7, each multicolumn is indexed by $k \in\{2,3,4, n\}$ and displays the following information: column "ILS" gives the lower bound obtained with the ILS metaheuristic; column "Primal best" presents the value of the best integer solution found in the search tree; column "(Gap)/Time" presents either the time (in seconds) spent to solve an instance to optimality or the final percentage gap (in brackets; between the best solution found and the upper bound) whenever the time limit is reached. The second column in these tables displays the primal bound $\alpha\left(G^{-}\right)$obtained. The last line in these tables informs us average values for the information at each column; for column "(Gap)/Time" the average gap is calculated on the set of non-solved instances while the average time is calculated on the set of instances solved to optimality.

The algorithm BC is able to solve more instances to optimality than $\mathrm{BB}+01$ and $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$; many of them in the root of the branch-and-cut tree. For both sets of instances and for each value of $k$, except for set 3.5 when $k=n$, the average gap of the non-solved instances is very small (less than 0.1 ). One can notice that, the average time to solve instances to optimality increases with $k$ for both set of instances. Results on these tables indicate that the $k$-MBS problem becomes numerically more difficult as $k$ increases and that instances in set 1.8 are slightly easier to be solved. Regarding the lower bounds at these tables, the quality of the lower bound provided by the ILS metaheuristic is slightly better than the one provided by $\alpha\left(G^{-}\right)$for $k \in\{2,3,4\}$. However, the lower bound provided by the ILS deteriorates as $k$ increases and, for $k=n$, it becomes worse when compared with $\alpha\left(G^{-}\right)$.

Additional tables ${ }^{2}$ exhibit information about the cut generation phase in the branch-and-cut method. We conclude that inequalities associated with cliques and 1-edge inequalities were fundamental for the solution of UNGA instances. Inequalities associated with holes and positive stars play a secondary role in the method but they are important for the solution of some instances. No inequality associated with holes was generated for instances in set 1.8 and this is explained by the fact that these instances presents higher density than instances 3.5. The average time spent in the separation phase is approximately $40 \%$ of the total time for instances 3.5 and varies, with $k$, from $51 \%$ to $70 \%$ for instances 1.8.

Finally, Table 8 displays a comparison of the three exact approaches implemented. On these tables, each multicolumn is indexed by $k \in\{2,3,4, n\}$ and by the type of method, "BB" or "BC" with the set of columns displaying the following information: " $R_{r}$ " is the value of the linear relaxation at the root of the search tree; " $\# 01$ " is the number of instances solved to optimality by $\mathrm{BB}+01$; "\#S1S2" is the number of instances solved to optimality by $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$; "\#BC" is the number of instances solved to optimality by BC. Each line of this table displays these information for a set of UNGA instances grouped according to the number of vertices, i.e., countries in the vote section. The last line of Table 8 exhibits either the average (for $R_{r}$ ) or the sum (for "\#01", "\#S1S2", "\#BC") of the information displayed at each column. Clearly, the performance of BC algorithm attests to the efficiency of the cuts and strategies proposed in Sections 3 and 4: not only the value of the linear relaxation at the root of the search three is better, but also more instances are solved to optimality.

Interpretation for the $55^{\text {th }}$ and $59^{\text {th }}$ Sessions. Coming back to Tables 6 and 7, we verify that for the majority of instances (UN sessions), $v(G, 3)=v(G, k)$, for each $4 \leq k \leq n$. We also notice a small distance from $v(G, 2)$ and $v(G, 3)$ except for instances 3.5 associated with sessions from the $55^{\text {th }}$ Session (2000-2001) to the $59^{\text {th }}$ Session (2004-2005). We do not present a complete interpretation of these results. Such a task would imply considerations out of the scope of the paper, e.g. parameters and methods used for the extraction of the vote

[^2]network, the analyses of networks associated with subjects discussed and voted in a given session (See [43]). However a brief analysis of the results obtained with the instance 3.5 associated with the $55^{\text {th }}$ Session, highlight the relevance of solving the $k$-MBS problem for different values of $k$. When comparing the optimal solution obtained when $k=2$ ( 170 countries) with the one obtained when $k=3$ (176 countries) (see Figure 8), we see that a set of countries leave the solution to give place to other ones:

- Leaving countries: Yugoslavia, St. Vincent and the Grenadines, Suriname and Gambia.
- Entering countries: Guinea-Bissau, Niger, Liberia, Central African Republic, Seychelles, Iraq, Jamaica, Sierra Leona, Togo and Angola.

Six countries, Guinea-Bissau, Niger, Liberia, Central African Republic, Seychelles, and Iraq, together with Mauritania compose the new group in the solution with $k=3$ while the other ones (among them Sierra Leona, Togo and Angola) joins the huge group of countries. The complete list of resolutions discussed in the $55^{\text {th }}$ Session can be found in [47] which includes 12 resolutions related with Africa subjects. The list [48] of the notable general assembly resolutions points Resolution A/RES/55/56 [1] from the $55^{t h}$ Session as a notable one. This resolution introduces the Kimberley Process Certification Scheme [49], a process to certify the origin of rough diamonds from sources that are conflict-free. The solution obtained when $k=3$ reflects the conflict around some resolutions which cannot be detected in the solution obtained with $k=2$.


Figure 4: Performance profile plots comparing the three methods on random instances, for each $k \in\{2,3, n\}$.

## 6. Conclusion

Signed graphs appear as an ideal discrete structure to represent relationships in collaborative vs. conflicting environments. Once the context of the application is defined, an optimal solution of the $k$-MBS problem can help us to gain a better understanding of the dynamics of this environment. To the best of our knowledge, the $k$-MBS problem has never been efficiently solved in the literature before. In this work, we introduced a representatives formulation for the $k$-MBS problem. With the purpose to develop an exact approach to its solution, we investigated some classes of valid inequalities for the associated polytope $P(G, k)$.


Figure 5: Performance profile plots comparing the three methods on random instances, for each $\varepsilon \in\{5,10,20\}$.

We have also proposed to use a constraint branching strategy based on a set of inequalities of the formulation. Extensive computational experiments were carried out over a set of random instances (with community structure) as well as over a set of instances representing voting on resolutions in the United Nations General Assembly on 63 sessions. The numerical results reported in Section 5, show the strength of some families of inequalities introduced and attests the efficiency of the branching strategy proposed. This later can be extended to other exact methods based on representatives formulation. Our current research focus on the evaluation of the optimal solution of the $k$-MBS problem as a criteria to measure the structural balance on real-world signed networks. In order to

Table 2: Number of random instances solved to optimality by each solution method.

|  |  | $n$ |  |  |  | $C$ |  |  |  |  |  |
| :--- | :--- | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| $k$ | Method | 60 | 70 | 80 | 2 | 3 | 4 | 5 | 5 | 10 | 20 |
| 2 | BB+01 | 18 | 1 | 0 | 6 | 4 | 5 | 4 | 16 | 3 | 0 |
|  | BB+S1S2 | 44 | 21 | 5 | 22 | 17 | 16 | 15 | 40 | 24 | 6 |
|  | BC | 60 | 45 | 10 | 29 | 26 | 30 | 30 | 47 | 38 | 30 |
| 3 | BB+01 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | BB+S1S2 | 16 | 2 | 0 | 6 | 6 | 3 | 3 | 17 | 1 | 0 |
|  | BC | 49 | 36 | 12 | 16 | 24 | 27 | 30 | 47 | 32 | 18 |
| 4 | BB+01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | BB+S1S2 | 7 | 2 | 0 | 3 | 4 | 2 | 1 | 10 | 0 | 0 |
|  | BC | 50 | 39 | 20 | 16 | 24 | 32 | 37 | 55 | 35 | 19 |
| $n$ | BB+01 | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 |
|  | BB+S1S2 | 3 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 |
|  | BC | 47 | 32 | 21 | 20 | 22 | 26 | 32 | 58 | 32 | 10 |

Table 3: Average gaps on random instances not solved to optimality by each solution method.

|  |  | $n$ |  |  |  |  | $C$ |  |  |  | $\varepsilon$ |  |  |
| :--- | :--- | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | :---: | :---: |
| $k$ | Method | 60 | 70 | 80 | 2 | 3 | 4 | 5 | 5 | 10 | 20 |  |  |
| 2 | BB+01 | 0.12 | 0.21 | 0.33 | 0.29 | 0.25 | 0.21 | 0.17 | 0.16 | 0.21 | 0.31 |  |  |
|  | BB+S1S2 | 0.06 | 0.13 | 0.24 | 0.21 | 0.18 | 0.17 | 0.15 | 0.11 | 0.15 | 0.21 |  |  |
|  | BC | - | 0.08 | 0.09 | 0.11 | 0.09 | 0.09 | 0.07 | 0.07 | 0.06 | 0.13 |  |  |
| 3 | BB+01 | 0.21 | 0.30 | 0.40 | 0.42 | 0.31 | 0.27 | 0.23 | 0.18 | 0.29 | 0.45 |  |  |
|  | BB+S1S2 | 0.15 | 0.21 | 0.33 | 0.31 | 0.25 | 0.21 | 0.18 | 0.13 | 0.20 | 0.35 |  |  |
|  | BC | 0.07 | 0.10 | 0.18 | 0.17 | 0.18 | 0.11 | 0.05 | 0.03 | 0.09 | 0.20 |  |  |
| 4 | BB+01 | 0.22 | 0.30 | 0.38 | 0.43 | 0.32 | 0.25 | 0.22 | 0.16 | 0.29 | 0.46 |  |  |
|  | BB+S1S2 | 0.19 | 0.26 | 0.34 | 0.35 | 0.29 | 0.24 | 0.20 | 0.13 | 0.24 | 0.40 |  |  |
|  | BC | 0.09 | 0.11 | 0.19 | 0.17 | 0.18 | 0.13 | 0.04 | 0.04 | 0.10 | 0.20 |  |  |
| $n$ | BB+01 | 0.36 | 0.52 | 0.74 | 0.79 | 0.62 | 0.43 | 0.33 | 0.44 | 0.54 | 0.64 |  |  |
|  | BB+S1S2 | 0.41 | 0.51 | 0.74 | 0.84 | 0.61 | 0.44 | 0.35 | 0.42 | 0.59 | 0.66 |  |  |
|  | BC | 0.09 | 0.23 | 0.44 | 0.49 | 0.34 | 0.18 | 0.11 | 0.33 | 0.32 | 0.31 |  |  |

be able to treat larger signed networks, efficient heuristic methods for the $k$ MBS problem must be developped. The ILS heuristic presented in this work performed quite well but for a subset of instances. Studying problem structures for which heuristic solutions are far from optimal solutions can drive us in the developpement of better heuristic strategies.

## Acknowledgements

This research benefited from the support of the "FMJH Program Gaspard Monge in optimization and operation research" (Project 2015-2842H), and from the support to this program from EDF. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

## References

[1] United Nations General Assembly Session 55 Resolution 56. The role of diamonds in fuelling conflict: breaking the link between the illicit transaction of rough diamonds and armed conflict as a contribution to prevention and settlement of conflicts A/RES/55/56. http://www.un.org/ga/search/


Figure 6: Linear relaxations and bounds obtained by BC on random instances when $k \in\{2,3\}$.


Figure 7: Percentage time and percentage of cuts per family obtained by BC on random instances when $k \in\{2,3,4,|V|\}$.
view_doc.asp?symbol=A/RES/55/56, December 2000. Accessed on March 2016.


Figure 8: Optimal solutions obtained for instance 3.5 associated with the $55^{t h}$ Session when $k=2$ (left) and $k=3$ (right).
[2] P. Abell and M. Ludwig. Structural balance: a dynamic perspective. Journal of Mathematical Sociology, 33:129-155, 2009.
[3] Z. Ales, A. Knippel, and A. Pauchet. Polyhedral combinatorics of the $k$ partitioning problem with representative variables. Discrete Applied Mathematics, 211:1-14, 2016.
[4] N. Arinik, R. Figueiredo, and V. Labatut. Signed graph analysis for the interpretation of voting behavior. In International Conference on Knowledge Technologies and Data-driven Business (i-KNOW), International Workshop on Social Network Analysis and Digital Humanities (SnanDig), Graz, Austria, 2017.
[5] A. Atamtürk, G.L. Nemhauser, and M. W.P. Savelsbergh. Conflict graphs in solving integer programming problems. European Journal of Operational Research, 121:40-55, Feb 2000.
[6] L. Bahiense, Y. Frota, N. Maculan, T.F. Noronha, and C.C. Ribeiro. A branch-and-cut algorithm for equitable coloring based on a formulation by representatives. Electronic Notes in Discrete Mathematics, 35:347-352, 2009.
[7] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In Proceedings of the 43 rd annual IEEE symposium of foundations of computer science, pages 238-250, Vancouver, Canada, 2002.
[8] F. Barahona and A.R. Mahjoub. Facets of the balanced (acyclic) induced subgraph polytope. Mathematical Programming, 45:21-33, 1989.
[9] J.J. Barthold. A good submatrix is hard to find. Operations Research Letters, 1:190-193, 1982.
[10] M. Campêlo, V.A. Campos, and R.C. Correa. On the asymmetric representatives formulation for the vertex coloring problem. Discrete Applied Mathematics, 156:1097-1111, 2008.
[11] M. Campelo, R.C. Correa, and Y. Frota. Cliques, holes and the vertex coloring polytope. Information Processing Letters, 89:1097-1111, 2004.
[12] D. Cartwright and F. Harary. Structural balance: A generalization of heiders theory. Psychological Review, 63:277-293, 1956.
[13] D. Cartwright and F. Harary. A note on a matrix criterion for unique colorability of a signed graph. Psychometrik, 32:291-296, 1967.
[14] D. Catanzaro, A. Godi, and M. Labbé. A class representative model for pure parsimony haplotyping. INFORMS Journal on Computing, 22:195209, 2010.
[15] M. Charikara, V. Guruswamib, and A. Wirtha. Clustering with qualitative information. Journal of Computer and System Sciences, 71:360-383, 2005.
[16] B. DasGupta, G. A. Encisob, E. Sontag, and Y. Zhanga. Algorithmic and complexity results for decompositions of biological networks into monotone subsystems. BioSystems, 90:161-178, 2007.
[17] J.A. Davis. Clustering and structural balance in graphs. Human Relations, 20:181-187, 1967.
[18] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. Theoretical Computer Science, 361:172187, 2006.
[19] P. Doreian and A. Mrvar. A partitioning approach to structural balance. Social Networks, 18:149-168, 1996.
[20] P. Doreian and A. Mrvar. Partitioning signed social networks. Social Networks, 31:1-11, 2009.
[21] L. Drummond, R. Figueiredo, Y. Frota, and M. Levorato. Efficient solution of the correlation clustering problem: an application to structural balance. Proceedings of OTM 2013 Workshops in LNCS, 8186:674-683, 2013.
[22] P. Esmailian, S. E. Abtahi, and M. Jalili. Mesoscopic analysis of online social networks: The role of negative ties. Phys. Rev. E, 90:042817, 2014.
[23] R. Figueiredo and Y. Frota. An improved branch-and-cut code for the maximum balanced subgraph of a signed graph. CoRR, 2013. arXiv/1312.4345.
[24] R. Figueiredo and Y. Frota. The maximum balanced subgraph of a signed graph: applications and solution approaches. European Journal of Operational Research, 236(2):473-487, 2014.
[25] R. Figueiredo, Y. Frota, and M. Labbé. Solution of the maximum $k$ balanced subgraph problem. In G. Nicosia and P. Pardalos, editors, Learning and Intelligent Optimization, pages 266-271, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
[26] R. Figueiredo, M. Labbé, and C.C. de Souza. An exact approach to the problem of extracting an embedded network matrix. Computers \& Operations Research, 38:1483-1492, 2011.
[27] R. Figueiredo and G. Moura. Mixed integer programming formulations for clustering problems related to structural balance. Social Networks, 35:639651, 2013.
[28] Y. Frota, N. Maculan, T. F. Noronha, and C. C. Ribeiro. A branch-and-cut algorithm for partition coloring. Networks, 55:194-204, 2010.
[29] M. R. Garey and D. S. Johnson. Computers and intractability. W.H. Freeman and Company, New York, 24 edition, 2003.
[30] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. In Proceedings of the National Academy of Sciences, volume 99, pages 7821-782, 2002.
[31] N. Gülpinar, G. Gutin, G. Mitra, and A. Zverovitch. Extracting pure network submatrices in linear programs using signed graphs. Discrete Applied Mathematics, 137:359-372, 2004.
[32] F. Harary and J.A. Kabell. A simple algorithm to detect balance in signed graphs. Mathematical Social Sciences, 1:131-136, 1980.
[33] F. Harary, M. Lim, and D. C. Wunsch. Signed graphs for portfolio analysis in risk management. IMA Journal of Management Mathematics, 13:1-10, 2003.
[34] F. Heider. Attitudes and cognitive organization. Journal of Psychology, 21:107-112, 1946.
[35] Padberg M. Hoffman, K.L. Solving airline crew scheduling problems. Management Sciense, 39:657-682, 1993.
[36] N. Howard. n-person soft games. Journal of the Operational Research Society, 49:144-150, 1998.
[37] T. Inohara. A formal theory on decision making with interperception. PhD thesis, Tokyo Institute of Technology, 1997.
[38] T. Inohara. On conditions for a meeting not to reach a deadlock. Applied Mathematics and Computation, 90:1-9, 1998.
[39] T. Inohara. Characterization of clusterability of signed graph in terms of newcomb's balance of sentiments. Applied Mathematics and Computation, 133:93-104, 2002.
[40] T. Inohara. Clusterability of groups and information exchange in group decision making with approval voting system. Applied Mathematics and Computation, 136:1-15, 2003.
[41] M. Levorato, R. Figueiredo, Y. Frota, and L. Drummond. Evaluating balancing on social networks through the efficient solution of correlation clustering problems. EURO Journal on Computational Optimization, 5:467498, 2017. doi:10.1007/s13675-017-0082-6.
[42] H.R. Lourenço, O.C. Martin, and T. Stutzle. Handbook of Metaheuristics, chapter Iterated Local Search, pages 1355-1377. Kluwer Academic Publishers, Norwell, MA, 2003.
[43] I. Mendonça, R. Figueiredo, V. Labatut, and P. Michelon. Relevance of negative links in graph partitioning: A case study using votes from the european parliament. In 2015 Second European Network Intelligence Conference, ENIC 2015, Karlskrona, Sweden, September 21-22, 2015, pages 122-129, 2015.
[44] N. Mladenović and P. Hansen. Variable neighborhood search. Computers § Operations Research, 24(11):1097-1100, 1997.
[45] G. L. Nemhauser and A. L. Wolsey. Integer and Combinatorial Optimization. Wiley-Interscience, New York, NY, USA, 1988.
[46] M.G.C. Resende and C.C. Ribeiro. Search Methodologies. Springer, 2nd edition edition, 2011.
[47] Resolutions and Decisions from the $55^{\text {th }}$ Session. Website. http:// research.un.org/en/docs/ga/quick/regular/55. Accessed on March 2016.
[48] The Notable General Assembly Resolutions. Website. https://en. wikipedia.org/wiki/United_Nations_General_Assembly_resolution. Accessed on March 2016.
[49] Kimberley Process Certification Scheme. Website. https://en. wikipedia.org/wiki/Kimberley_Process_Certification_Scheme. Accessed on March 2016.
[50] B. Yang, W.K. Cheung, and J. Liu. Community mining from signed social networks. IEEE Transactions on Knowledge and Data Engineering, 19:1333-1348, 2007.
[51] T. Zaslavsky. A mathematical bibliography of signed and gain graphs and allied areas. Electronic Journal of Combinatorics DS8, 1998.

Table 4: Description of UNGA instances extracted with the algorithm described in [24] by setting $\alpha=0.3, \beta=0.5$ and $\alpha=0.1, \beta=0.8$.

| Name | $n$ | 3.5 |  |  |  | 1.8 |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $m^{+}$ | $m^{-}$ | $d^{+}$ | $d^{-}$ | $m^{+}$ | $m^{-}$ | $d^{+}$ | $d^{-}$ |
| S01 | 54 | 663 | 269 | 0,46 | 0,19 | 1064 | 451 | 0,74 | 0,32 |
| S02 | 57 | 616 | 246 | 0,39 | 0,15 | 1040 | 662 | 0,65 | 0,41 |
| S03 | 59 | 717 | 269 | 0,42 | 0,16 | 1372 | 630 | 0,80 | 0,37 |
| S04 | 59 | 793 | 177 | 0,46 | 0,10 | 1405 | 520 | 0,82 | 0,30 |
| S05 | 60 | 1009 | 144 | 0,57 | 0,08 | 1503 | 424 | 0,85 | 0,24 |
| S06 | 60 | 789 | 228 | 0,45 | 0,13 | 1321 | 650 | 0,75 | 0,37 |
| S07 | 60 | 1049 | 102 | 0,59 | 0,06 | 1541 | 424 | 0,87 | 0,24 |
| S08 | 60 | 983 | 267 | 0,56 | 0,15 | 1394 | 494 | 0,79 | 0,28 |
| S09 | 60 | 869 | 64 | 0,49 | 0,04 | 1510 | 486 | 0,85 | 0,27 |
| S10 | 65 | 1551 | 403 | 0,75 | 0,19 | 1880 | 466 | 0,90 | 0,22 |
| S11 | 81 | 2132 | 438 | 0,66 | 0,14 | 2582 | 906 | 0,80 | 0,28 |
| S12 | 82 | 1805 | 385 | 0,54 | 0,12 | 2647 | 985 | 0,80 | 0,30 |
| S13 | 82 | 2191 | 295 | 0,66 | 0,09 | 2829 | 628 | 0,85 | 0,19 |
| S14 | 82 | 1854 | 366 | 0,56 | 0,11 | 2768 | 984 | 0,83 | 0,30 |
| S15 | 100 | 2319 | 577 | 0,47 | 0,12 | 3962 | 1905 | 0,80 | 0,38 |
| S16 | 106 | 2858 | 915 | 0,51 | 0,16 | 4512 | 2246 | 0,81 | 0,40 |
| S17 | 110 | 2800 | 1026 | 0,47 | 0,17 | 4528 | 2532 | 0,76 | 0,42 |
| S18 | 113 | 4053 | 549 | 0,64 | 0,09 | 5297 | 1261 | 0,84 | 0,20 |
| S20 | 117 | 3629 | 560 | 0,53 | 0,08 | 5566 | 1774 | 0,82 | 0,26 |
| S21 | 122 | 3967 | 854 | 0,54 | 0,12 | 6166 | 2293 | 0,84 | 0,31 |
| S22 | 124 | 3613 | 954 | 0,47 | 0,13 | 5704 | 2488 | 0,75 | 0,33 |
| S23 | 126 | 3731 | 784 | 0,47 | 0,10 | 6554 | 2411 | 0,83 | 0,31 |
| S24 | 126 | 3494 | 699 | 0,44 | 0,09 | 6276 | 2359 | 0,80 | 0,30 |
| S25 | 127 | 3438 | 718 | 0,43 | 0,09 | 6092 | 2442 | 0,76 | 0,31 |
| S26 | 133 | 5080 | 666 | 0,58 | 0,08 | 7393 | 1737 | 0,84 | 0,20 |
| S27 | 132 | 6519 | 133 | 0,75 | 0,02 | 8184 | 610 | 0,95 | 0,07 |
| S28 | 135 | 6615 | 464 | 0,73 | 0,05 | 8213 | 1161 | 0,91 | 0,13 |
| S29 | 138 | 7111 | 487 | 0,75 | 0,05 | 8843 | 957 | 0,94 | 0,10 |
| S30 | 143 | 7364 | 1268 | 0,73 | 0,12 | 8895 | 1907 | 0,88 | 0,19 |
| S31 | 144 | 8434 | 1462 | 0,82 | 0,14 | 9449 | 1766 | 0,92 | 0,17 |
| S32 | 146 | 9008 | 937 | 0,85 | 0,09 | 9993 | 1309 | 0,94 | 0,12 |
| S33 | 148 | 8324 | 850 | 0,77 | 0,08 | 10134 | 1385 | 0,93 | 0,13 |
| S34 | 150 | 8813 | 1031 | 0,79 | 0,09 | 10341 | 1851 | 0,93 | 0,17 |
| S35 | 151 | 8542 | 874 | 0,75 | 0,08 | 10395 | 1804 | 0,92 | 0,16 |
| S36 | 155 | 8849 | 978 | 0,74 | 0,08 | 10756 | 1809 | 0,90 | 0,15 |
| S37 | 156 | 9629 | 1952 | 0,80 | 0,16 | 11213 | 2598 | 0,93 | 0,21 |
| S38 | 157 | 9134 | 1201 | 0,75 | 0,10 | 10750 | 2171 | 0,88 | 0,18 |
| S39 | 158 | 9867 | 2236 | 0,80 | 0,18 | 11246 | 2995 | 0,91 | 0,24 |
| S40 | 158 | 9962 | 1942 | 0,80 | 0,16 | 11733 | 2903 | 0,95 | 0,23 |
| S41 | 158 | 10031 | 305 | 0,81 | 0,02 | 11715 | 1110 | 0,94 | 0,09 |
| S42 | 158 | 10884 | 289 | 0,88 | 0,02 | 12039 | 574 | 0,97 | 0,05 |
| S43 | 158 | 10357 | 330 | 0,84 | 0,03 | 11792 | 954 | 0,95 | 0,08 |
| S44 | 158 | 10255 | 301 | 0,83 | 0,02 | 11552 | 881 | 0,93 | 0,07 |
| S45 | 158 | 10040 | 900 | 0,81 | 0,07 | 11115 | 1463 | 0,90 | 0,12 |
| S46 | 178 | 12322 | 2753 | 0,78 | 0,17 | 14683 | 3518 | 0,93 | 0,22 |
| S47 | 180 | 10059 | 1972 | 0,62 | 0,12 | 12977 | 3747 | 0,81 | 0,23 |
| S48 | 184 | 10680 | 2059 | 0,63 | 0,12 | 13478 | 3563 | 0,80 | 0,21 |
| S49 | 185 | 11512 | 2568 | 0,68 | 0,15 | 13625 | 3383 | 0,80 | 0,20 |
| S50 | 185 | 10633 | 3049 | 0,62 | 0,18 | 12969 | 4472 | 0,76 | 0,26 |
| S51 | 185 | 12078 | 1990 | 0,71 | 0,12 | 14087 | 2993 | 0,83 | 0,18 |
| S52 | 176 | 11685 | 1597 | 0,76 | 0,10 | 13994 | 2753 | 0,91 | 0,18 |
| S53 | 177 | 11819 | 1043 | 0,76 | 0,07 | 14057 | 1885 | 0,90 | 0,12 |
| S54 | 182 | 11320 | 2692 | 0,69 | 0,16 | 13155 | 3877 | 0,80 | 0,24 |
| S55 | 189 | 11228 | 3057 | 0,63 | 0,17 | 13018 | 4404 | 0,73 | 0,25 |
| S56 | 191 | 9659 | 4304 | 0,53 | 0,24 | 12345 | 5754 | 0,68 | 0,32 |
| S57 | 192 | 11648 | 2703 | 0,64 | 0,15 | 14362 | 4307 | 0,78 | 0,23 |
| S58 | 191 | 12830 | 2047 | 0,71 | 0,11 | 15384 | 3140 | 0,85 | 0,17 |
| S59 | 191 | 13572 | 967 | 0,75 | 0,05 | 16374 | 1879 | 0,90 | 0,10 |
| S60 | 192 | 12855 | 2064 | 0,70 | 0,11 | 15177 | 3407 | 0,83 | 0,19 |
| S61 | 192 | 13272 | 1055 | 0,72 | 0,06 | 15728 | 2532 | 0,86 | 0,14 |
| S62 | 192 | 12267 | 848 | 0,67 | 0,05 | 15671 | 2510 | 0,85 | 0,14 |
| S63 | 192 | 13712 | 1540 | 0,75 | 0,08 | 15788 | 2569 | 0,86 | 0,14 |
| Avg. | 136.13 | 7078.90 | 1100.05 | 0.65 | 0.11 | 8744.13 | 1984.82 | 0.85 | 0.22 |

Table 5: Branch-and-bound results on UNGA instances.

|  |  | 1.8 |  |  |  |  | 3.5 |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $k$ | Method | Primal ${ }_{\text {best }}$ | Gap \% | Time | Nodes | \#Solved | Primal $_{\text {best }}$ | Gap \% | Time | Nodes | \#Solved |
| 2 | BB+01 | 101.05 | 0.22 | 2145.35 | 71.29 | 7 | 118.34 | 0.14 | 497.61 | 44.50 | 7 |
|  | BB+S1S2 | 100.98 | 0.20 | 346.72 | 8.10 | 9 | 118.11 | 0.13 | 282.48 | 10.00 | 10 |
| 3 | BB+01 | 101.71 | 0.32 | 2849.98 | 66.34 | 2 | 119.15 | 0.18 | 1306.60 | 42.34 | 5 |
|  | $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ | 102.07 | 0.26 | 940.72 | 14.85 | 10 | 117.74 | 0.17 | 1011.09 | 30.06 | 9 |
| 4 | BB+01 | 100.45 | 0.39 | - | 104.28 | 0 | 114.73 | 0.26 | 2103.98 | 78.23 | 4 |
|  | $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ | 101.20 | 0.30 | 978.04 | 18.46 | 10 | 115.13 | 0.23 | 1540.11 | 49.31 | 10 |
| $n$ | $\mathrm{BB}+01$ | 75.11 | 1.00 | 1151.70 | 656.98 | 3 | 65.69 | 1.41 | 1653.77 | 370.53 | 6 |
|  | $\mathrm{BB}+\mathrm{S} 1 \mathrm{~S} 2$ | 73.18 | 1.18 | 364.24 | 41.51 | 6 | 60.48 | 1.76 | 437.97 | 94.93 | 6 |


















 8




[^0]:    * Corresponding author.

[^1]:    ${ }^{1}$ Notice that, in [17, 19], the definition of a $k$-balanced signed graph is slightly different since $k$ is not considered to be a fixed parameter.

[^2]:    ${ }^{2}$ Available in http://www.ic.uff.br/~yuri/files/kmbs.zip.

