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

Rosa Figueiredo, Yuri Frota, Martine Labbé

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

Rosa Figueiredo[a,*], Yuri Frota[b], Martine Labbé[c]

[a]*Laboratoire Informatique d'Avignon, University of Avignon*
84911 Avignon cedex 9, France.
rosa.figueiredo@univ-avignon.fr
[b]*Department of Computer Science, Fluminense Federal University*
24210-240 Niterói-RJ, Brazil.
yuri@ic.uff.br
[c]*Université Libre de Bruxelles, Département d'Informatique*
CP 210/01 B-1050 Brussels, Belgium.
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 *vs* 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.

*Corresponding author.

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 $s : E \to \{+, -\}$ 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 $V \setminus W$ in such a way that $E[W] \cup E[V \setminus W] = E^+$. 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 $\cup_{1 \leq i \leq l} E[N_i] = E^+$. 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 $H = (V', E[V'], s)$of G such that H is k-balanced and maximizes the cardinality of the vertex set V'. 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]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.

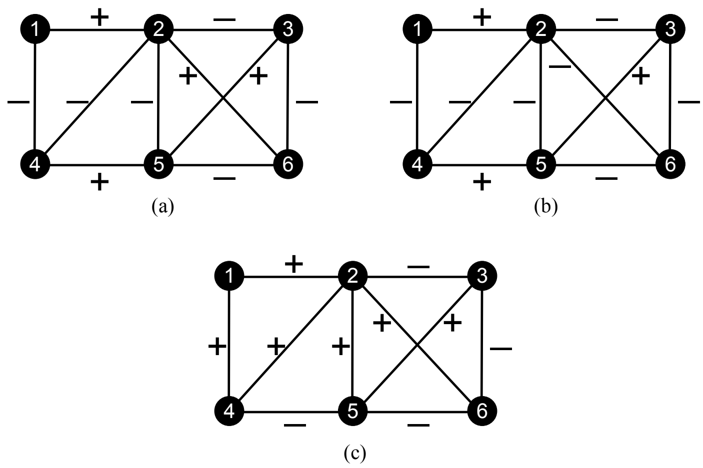

Figure 1: (a) A balanced signed graph: $W = \{1, 2, 6\}$ and $V \setminus W = \{3, 4, 5\}$. (b) A k-balanced signed graph for any $k \in \{3, 4, 5, 6\}$. Sets $N_1 = \{1, 2\}$, $N_2 = \{3, 4, 5\}$ and $N_3 = \{6\}$ define a 3-partition of this graph. (c) A non-clusterable signed graph.

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 solution 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 $s : E \to \{+, -\}$ 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 $G^- = (V, E^-)$ and $G^+ = (V, E^+)$. For a vertex set $S \subseteq V$, we define $N^-(S) = \{j \in V \mid (i, j) \in E^-, i \in S\}$, $N^+(S) = \{j \in V \mid (i, j) \in E^+, i \in S\}$ and $\bar{N}^-(S) = V \setminus (N^-(S) \cup S)$. 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 π (property π 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], pp. 195). For the k-MBS problem, property π 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. $\alpha(G^-) = v(G, 1) \leq v(G, 2) \leq \ldots \leq v(G, n)$.

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 $H = (V^H, E^H)$ of G which implies $E[V^H] = E^H \subseteq E^+$ and V^H is an independent set in G^-. A feasible solution for the independent set problem defined over $G^- = (V, E^-)$ 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 correlation-clustering 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 $A = \{(i, j) \mid i \in V, j \in \bar{N}^-(\{i\}), i < j\}$ 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 $D(i) = \{j \in V \mid (i, j) \in A^0\}$ denote the set of all vertices that can be represented by i and let $O(j) = \{i \in V \mid (i, j) \in A^0\}$ 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 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.

$$\text{Maximize} \sum_{(i,j)\in A^0} x_j^i \tag{1}$$

$$\text{subject to} \sum_{i\in O(j)} x_j^i \leq 1, \qquad \forall\ j \in V, \tag{2}$$

$$\sum_{i\in V} x_i^i \leq k, \tag{3}$$

$$x_j^i \leq x_i^i, \qquad \forall\ (i,j) \in A, \tag{4}$$

$$x_i^p + x_j^p \leq x_p^p, \qquad \forall\ (i,j) \in E^-, \forall\ p \in O(i) \cap O(j), \tag{5}$$

$$x_i^p + x_j^q \leq 1, \qquad \forall\ (i,j) \in E^+, \forall\ p \in O(i), \forall\ q \in O(j) \setminus \{p\}, \tag{6}$$

$$x_i^i \in \{0,1\}, \qquad \forall\ i \in V, \tag{7}$$

$$x_j^i \in \{0,1\}, \qquad \forall\ (i,j) \in A. \tag{8}$$

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 $IP(G, k)$.

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 $IP(G, k)$ is defined here as

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

Additional notations will be necessary before we can proceed. Consider a set of l disjoint vertex sets $I = \{N_1, \ldots, N_l\}$. 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 $I_{ij} = \{\{i,j\}\}$, for $(i,j) \in A$.

Lemma 3.1. *The polytope $P_{G,k}$ is full-dimensional, i.e.,* $\dim(P_{G,k}) = n + m$.

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 $x^{I_{ij}}$, 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 $IP(G,k)$ 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 $F = \{x \in P_{G,k} \mid b^T x = \beta\}$ 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 $F \subseteq F_a = \{x \in P_{G,k} \mid a^T x = \alpha\}$ 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 $i \in V \setminus \{p\}$, $(i,p) \notin E^+$.

Proof. Suppose that $a_p^p = \alpha$ and let $i \in O(p) \setminus \{p\}$. Define the set of disjoint vertex sets $I_1 = \{\{p\},\{i\}\}$. From the incidence vectors x^{I_1} and $x^{I_{ip}}$ in $a^T x = \alpha$ we can conclude that $a_p^i = \alpha$. Consider an arc $(p,j) \in A$. The solutions x^{I_p} and $x^{I_{pj}}$ 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 $a^i_j = 0$. Consider a vertex $i \in V \setminus \{p\}$. 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 $\sum_{i \in O(p)} x^i_p + x^q_q \leq 1$, if $q > p$, and by inequality $\sum_{i \in O(p) \setminus \{q\}} x^i_p + x^q_q \leq 1$, if $q < p$. □

Lemma 3.4. *Consider an arc $(p, q) \in A$. Inequality (4) written for (p, q), i.e.,*

$$x^p_q - x^p_p \leq 0,$$

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 $x^{I_{pq}}$ and the null vector in $a^T x = \alpha$ lead us to $a^p_q = -a^p_p$. Consider a vertex $i \in V \setminus \{p\}$. 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 $x^{I_{ij}}$ and the null vector we arrive to $a^i_j = -a^i_i = 0$. Now consider an 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 $x^{I_{pq}}$ and x^{I_2} we conclude that $a^p_j = 0$. 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, $x^p_q + x^p_j \leq x^p_p$. □

Constraints (6) can be replaced in formulation $IP(G, k)$ 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*

$$\sum_{p \in S} x^p_i + \sum_{p \in O(j) \setminus S} x^p_j \leq 1, \tag{9}$$

is valid for $P_{G,k}$ and dominates constraints (6) in formulation $IP(G, k)$.

Proof. Consider a vertex $p \in S$ and let $T = O(j) \setminus \{p\}$. Adding inequalities (6) written to each combination of vertices p and $q \in T$, and dividing the result by $|T|$ we obtain the inequality $x^p_i + (1/|T|) \sum_{q \in T} x^p_j \leq 1$. Since $\sum_{q \in T} x^p_j \in \{0, 1\}$, the fractional coefficient in the left-hand side of this inequality can be rounding up giving us $x^p_i + \sum_{q \in T} x^p_j \leq 1$. 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*

$$\sum_{i \in K^-} x_i^p \leq x_p^p, \tag{10}$$

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 $|K^-| = 1$, 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 $|K^-| = p + 1$. Consider any two vertices $u, v \in K^-$. The following inequalities are valid: inequality (10) written to $K^- \setminus \{u\}$; inequality (10) written to $K^- \setminus \{v\}$; 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 $x^{I_{pi}}$ 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 $i \in V \setminus \{p\}$. From solution x^{I_i} and the null vector we arrive to $a_i^i = 0$. Consider an arc $(j, i) \in A$ with $i \in K^-$ and $j \in V \setminus \{p\}$. From the incidence vector $x^{I_{ji}}$ 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 $j \in V \setminus \{p\}$. From the incidence vector $x^{I_{ij}}$ and the null vector, we conclude that $a_j^i = 0$. Consider an arc $(p, j) \in A$ with $j \in V \setminus K^-$. 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 $x^{I'}$ with $I' = \{p, j, i\}$. The incidence vectors $x^{I'}$ and $x^{I_{pi}}$ in $a^T x = \alpha$ lead us to $a_j^p = 0$. Now consider an arc (j, p) with $j \in V \setminus K^-$. In that case, from the incidence vector $x^{I_{jp}}$ and the null vector, we conclude that $a_p^j = 0$. Let $(i, j) \in A$ be an arc such that $i, j \in V \setminus K^- \cup \{p\}$. From solution $x^{I_{ij}}$ and the null vector, we arrive to $a_j^i = 0$. Finally, assume that there is a vertex $j \in V \setminus K^- \cup \{p\}$ 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 $S(j) \subseteq O(j)$. Additionally, assume* *that $S(i) \cap S(j) = \emptyset$ for each pair $i, j \in K^+$. The inequality*

$$\sum_{j \in K^+} \sum_{i \in S(j)} x_j^i \leq 1, \tag{11}$$

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., $O(j) \setminus \cup_{p \in K} S(p) = \emptyset$,*

(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 $a \in \Re^{n+m=|A \cup A^0|}$. 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, $S(K^+) = \cup_{p \in K} S(p) = \emptyset$ and, for each $i \in S(K^+)$, 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 $S(j) \subseteq O(j) \setminus \{p\}$. Additionally, assume that $S(i) \cap S(j) = \emptyset$ for each pair $i, j \in K^+$. The inequality*

$$\sum_{i \in K^-} x_i^p + \sum_{j \in K^+} \sum_{i \in S(j)} x_j^i \leq 1, \tag{12}$$

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

Proof. We use induction on the length of negative clique K^-. If $|K^-| = 1$, 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 $I_{i'}, i' \in K^+$	$a_i^i = \alpha$	
$(i,i), i \in S(K^+)$	I_p $\{\{i\},\{p\}\}$	$a_i^i = 0$	$\exists p \in K^+$ s.t. $(i,p) \notin E^+$ due to (i)
$(i,i), i \in V \setminus (K^+ \cup S(K^+))$	I_p $\{\{i\},\{p\}\}$	$a_i^i = 0$	$\exists p \in K^+$ s.t. $(i,p) \notin E^+$ due to (i)
$(i,j), j \in K^+, i \in S(j)$	I_{ij}, I_j	$a_j^i = \alpha$	
$(i,j), j \in K^+, i \in S(K^+ \setminus j)$	$\{\{i,j,p_i\}\}$ I_{ip_i}	$a_j^i = 0$	
$(i,j), j \in K^+, i \in V \setminus K^+ \setminus S(K^+)$			(i,j) do not exist due to (ii)
$(i,j), j \in S(K^+), i \in K^+$	I_{ij}, I_i	$a_j^i = 0$	
$(i,j), j \in S(K^+), i \in S(K^+)$	$\{\{i,j\},\{p\}\}$ I_p	$a_j^i = 0$	p exists due to (iii)
$(i,j), j \in S(K^+), i \in V \setminus K^+ \setminus S(K^+)$	$\{\{i,j\},\{p\}\}$ I_p	$a_j^i = 0$	p exists due to (iii)
$(i,j), i \in K^+, j \in V \setminus K^+ \setminus S(K^+)$	I_{ij}, I_i	$a_j^i = 0$	
$(i,j), i \in S(K^+), j \in V \setminus K^+ \setminus S(K^+)$	$\{\{i,j\},\{p\}\}$ I_p	$a_j^i = 0$	p exists due to (iii)
$(i,j), i,j \in V \setminus K^+ \setminus S(K^+)$	$\{\{i,j\},\{p\}\}$ I_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.

$|K^-| = p + 1$. Consider a vertex $u \in K^-$. The following inequalities are valid: $\sum_{i \in K^- \setminus \{u\}} x_i^p + \sum_{j \in K^+} \sum_{i \in S(j)} x_j^i \leq 1$ 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*

$$\sum_{i \in H} x_i^p \leq \lfloor (|H| + t)/2 \rfloor \, x_p^p, \tag{13}$$

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 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 arc $(i,j) \in E^-$ such that $i,j \in H$, and the inequalities $x_i^p + x_j^p \leq 2x_p^p$ 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*

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

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)* $(j^-, j), (j, j^+) \in E^+$ *and (ii)* $j > j^-, j^+$; $a_j^j = 1$ *otherwise. The inequality*

$$\sum_{j \in C} \left(a_j^j x_j^j + \sum_{i \in O(j) \setminus \{j\}} x_j^i \right) \leq |C| - 1, \tag{15}$$

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

Proof. By definition, a k-balanced signed graph is clusterable. Hence, the following inequality is clearly valid:

$$\sum_{j \in C} \left(x_j^j + \sum_{i \in O(j) \setminus \{j\}} x_j^i \right) \leq |C| - 1. \tag{16}$$

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:

$$\alpha_p^p x_p^p + \sum_{i \in O(p) \setminus \{p\}} x_p^i + \sum_{j \in C \setminus \{p\}} \left(x_j^j + \sum_{i \in O(j) \setminus \{j\}} x_j^i \right) \leq |C| - 1. \tag{17}$$

The assignment $x_p^p = 0$ do not impose a limit to the value of α_p^p. However,

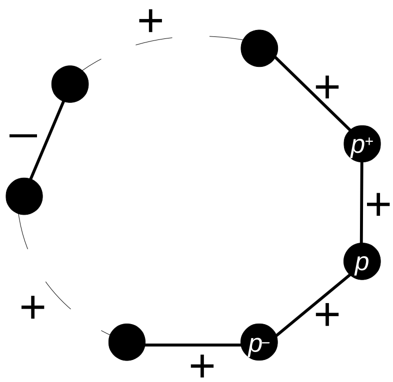

Figure 2: A cycle C with exactly one negative edge: at most $|C|-1$ vertices can be simultaneously in a feasible solution. If vertex p is a representative one, vertices p^- and p^+ are prevented to be in the solution.

whenever $x_p^p = 1$,

$$\alpha_p^p \leq (|C|-1) - \sum_{i \in O(p)\setminus\{p\}} x_p^i - \sum_{j \in C\setminus\{p\}} \left(x_j^j + \sum_{i \in O(j)\setminus\{j\}} x_j^i \right).$$

If p is a representative vertex ($x_p^p = 1$) 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 $C \setminus \{p, p^-, p^+\}$ 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). $\square$

A *star* in G is a tree $X_l = (V^X, E^X)$ 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 $X_l = (V^X, E^X)$ be a positive star in G with internal node p such that $k \geq l$ and, for each $j \in V^X \setminus \{p\}$, $p > j$. For each vertex $j \in V^X \setminus \{p\}$, consider a set $S(j) \subseteq O(j)$. Additionally, assume that $S(i) \cap S(j) = \emptyset$ for each pair $i, j \in V^X \setminus \{p\}$. The inequality*

$$\sum_{j \in V^X \setminus\{p\}} \left((k-1)x_p^j + \sum_{i \in S(j)} x_j^i \right) + kx_p^p \leq k, \tag{18}$$

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

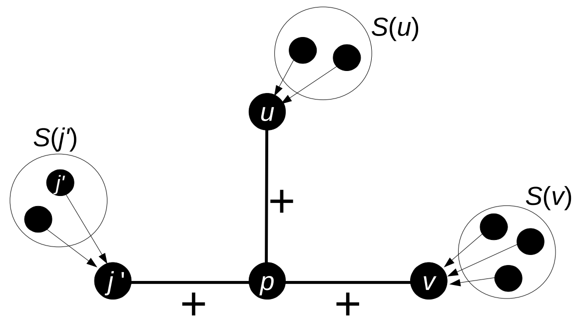

Figure 3: A positive star $X_3 = (V^X, E^X)$ with internal node p such that $k \geq 3$. The leafs of the positive star cannot belong to the solution whenever p is a representative vertex. On the other hand, j' can represent p and in that case any other leaf of the positive star is prevented to be in the feasible solution.

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

$$\sum_{j \in V^X \setminus \{p\}} (k-1) x_p^j + k x_p^p \leq k. \tag{19}$$

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

$$\sum_{j \in V^X \setminus \{p\}} \sum_{i \in S(j)} x_j^i \leq k. \tag{20}$$

Since X_l is a positive star, the left-hand side of (19) equal to k ($x_p^p = 1$) 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;$x_p^{j'} = 1, j' \in V^X \setminus \{p\}$) implies the left-hand side of (20) less than or equal to one (being one whenever $j' \in S(j')$). □

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 $(V_1, \ldots, V_k)$ such that $V_1, \ldots, V_k \subseteq V$, $\bigcup_{i=1,\ldots,k} E[V_i] \subseteq E^+$ and $(E[V_1 \cup \ldots \cup V_k]) \backslash (\bigcup_{i=1,\ldots,k} E[V_i]) \subseteq E^-$.

Construction phase. The overall method attempts to find a maximal feasible solution $(V_1, \ldots, V_k)$. 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 $(V_1, \ldots, V_k)$ be a feasible solution. We define $Cand(V_i) = \{u \in V \setminus \cup_{i=1,\ldots,k} V_i \mid (V_1, \ldots, V_i \cup \{u\}, \ldots, V_k)$ is a feasible solution$\}$, the set of vertices that are candidate to be added to set V_i. At each iteration, a vertex $u \in Cand(V_i)$ (for $i = 1, \ldots, k$) is randomly selected and a new set V_i' is reached by inserting vertex u into the set V_i. The procedure is repeated until $Cand(V_i) = \emptyset$, 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 $v \in (Cand(V_1) \cup \ldots \cup Cand(V_k))$ into the corresponding subset (i.e., V_1 if $v \in Cand(V_1)$, V_2 if $v \in Cand(V_2)$, ...). The process is repeated until $(Cand(V_1) \cup \ldots \cup Cand(V_k)) = \emptyset$.

Drop(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 $V_j = V_j \cup \{u\}$) and **Hire**.

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

2Drop(i)-Hire: removing two vertex u_1 and u_2 from V_i (i.e., $V_i = V_i \backslash \{u_1, u_2\}$) 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' (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'; 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 breadth-first 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 $D^a = (V_1^a \cup V_2^a, A^a)$ defined next. For each $i \in V$, we create a vertice $i \in V_1^a$ and a vertice $i' \in V_2^a$. For each positive edge $(i,j) \in E^+$, we create four arcs $(i,j), (j,i), (i',j'), (j',i') \in A^a$ while for each negative edge $(i,j) \in E^-$, we create two arcs $(i,j'), (j,i') \in A^a$. A length $d_{ij} = 1 - z_i$ is defined for each 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 $i` \in V_2^a$ in $D^a$. Notice that, each path from $i \in V_1^a$ to $i` \in V_2^a$ 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 $X_l = (V^X, E^X)$ with internal vertex p. Even the problem of deciding if there exist, for each leaf $j \in V^X \setminus \{p\}$, 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 $H = (V^H, E^H)$ described next. For each $i \in S(j)$ and for each $j \in V^X \setminus \{p\}$, create a vertex ij in V^H. For each pair of vertices $ij_1, ij_2 \in V^H$ create an edge $(ij_1, ij_2) \in$

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 $H = (V^H, E^H)$. 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_{ij}, $(i,j) \in A^0$, 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 $(\mid V \mid /10)$ 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 $j \in V \setminus p$. First, we define a partition S_1, S_2 for the set $O(p)$ such that $\sum_{j \in S1} x_p^j$ (and $\sum_{j \in S1} \bar{x}_p^j$) is a fractional value. Sets S_1 and S_2 are defined in a way that $| \sum_{j \in S1} x_p^j - \sum_{j \in S2} x_p^j|$ is minimized. We create three branches on the branching tree by adding constraints defined on sets S_1 and S_2:

$\sum_{j \in S1} x_p^j = 1$, p is in the solution and is represented by a vertex in S_1,

$\sum_{j \in S2} x_p^j = 1$, 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 $SG(c, n, p_-, p_+, \epsilon)$ 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 $c \times \frac{n}{2c}(\frac{n}{c} - 1) \times p_+$ positive links internal to the c communities and $\frac{c(c-1)}{2} \times \frac{n}{2c}\frac{n}{c} \times p_-$ negative links among the c communities, with each vertice having around $(\frac{n}{c} - 1) \times p_+ + (\frac{n}{c} - 1) \times p_+$ 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 ϵ. The parameters c, n and ϵ 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, α and β, 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 $d^+ = 2m^+/n(n-1)$ of the network, the number of negative edges "m^-" and the negative density $d^- = 2m^-/n(n-1)$ 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.

- 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 $\alpha(G^-)$ 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 1h 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 BB+01, BB+S1S2 and the branch-and-cut algorithm BC for $k \in \{2, 3, 4, |V|\}$.

5.3.1. Random instances

BB methods vs BC. 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 ε. 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 BB+01 method remains the same as k increases while the performance of BB+S1S2 method deteriorates, practically equating to BB+01 method when $k = n$. From 5, we arrive to the same conclusion as ε increases. Clearly, the BC method obtained the best performance among the three methods. We can also observe that, the performance of BB+01 and BB+S1S2 are closer to the performance of BC method as parameter ε 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 ϵ. From these tables, we conclude that BB+S1S2 has slightly better performance than 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. BB+01 and BB+S1S2) 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, BB+01 and 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, ϵ and k, than BB+01 and BB+S1S2.

BC 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

BB+01 vs BB+S1S2. Table 5 displays results obtained with BB+01 and BB+S1S2 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: "$Primal_{best}$" 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).

BC 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 $\alpha(G^-)$ 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 BB+01 and BB+S1S2; 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 $\alpha(G^-)$ 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 $\alpha(G^-)$.

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 BB+01; "#S1S2" is the number of instances solved to optimality by BB+S1S2; "#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^{th} *and* 59^{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^{th} Session (2000–2001) to the 59^{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] Available in `http://www.ic.uff.br/~yuri/files/kmbs.zip`.

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^{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^{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^{th} 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 $k = 2$.

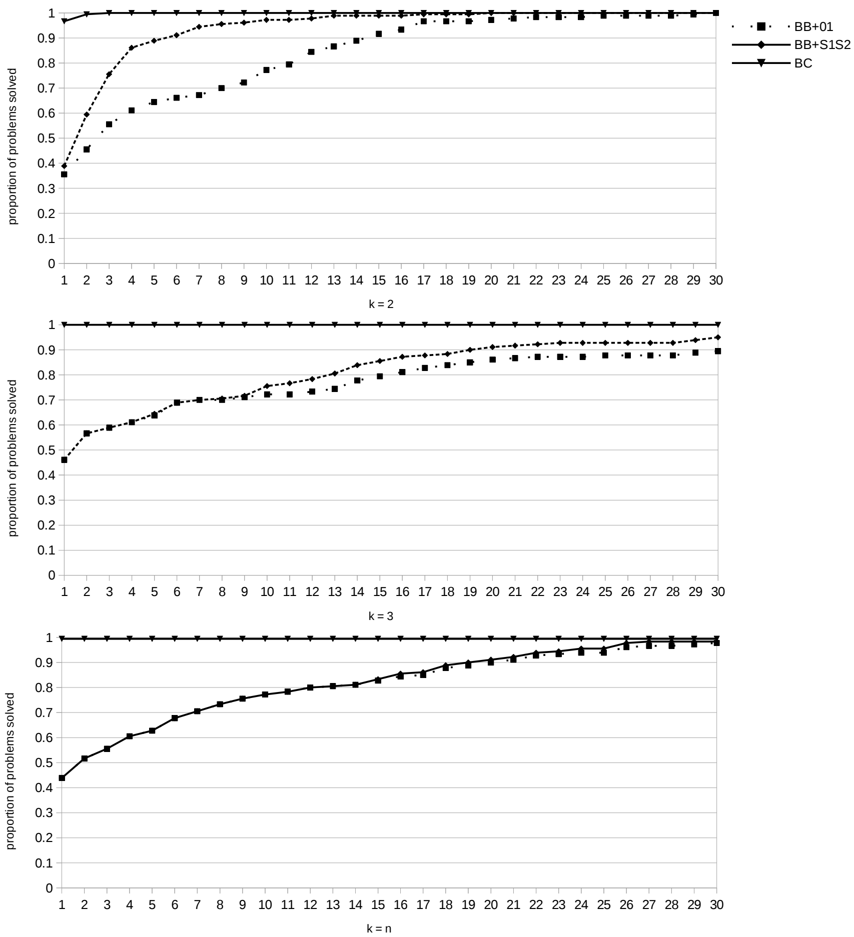

Figure 4: Performance profile plots comparing the three methods on random instances, for each $k \in \{2, 3, n\}$.

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 $P(G, k)$.

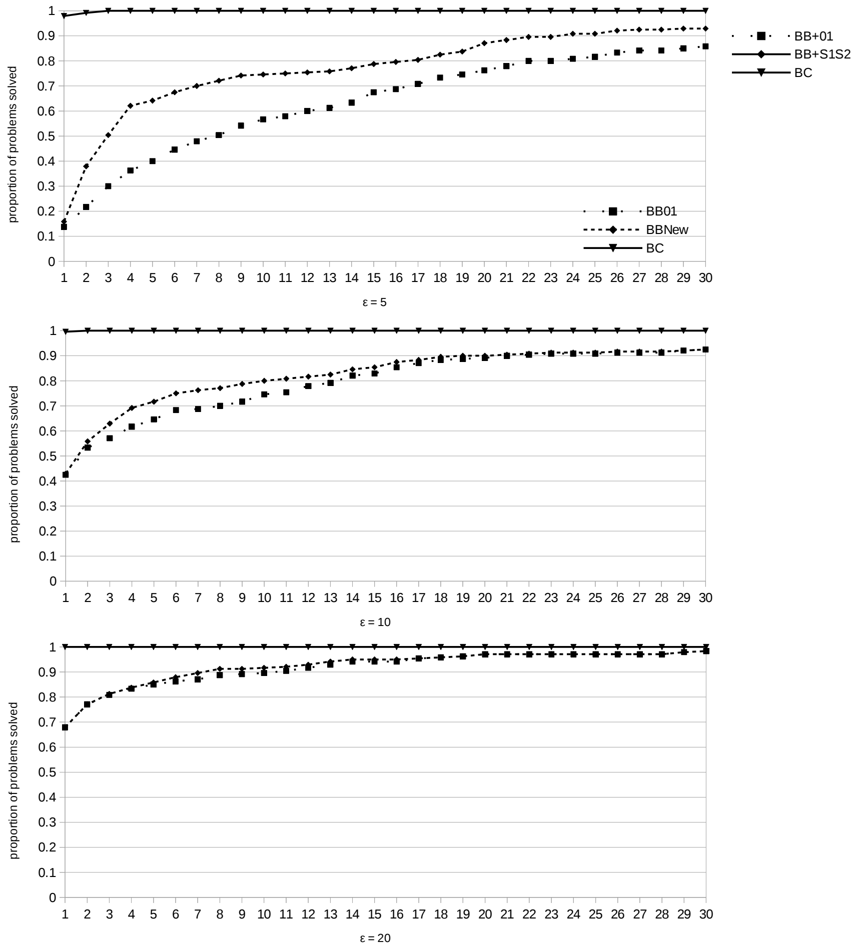

Figure 5: Performance profile plots comparing the three methods on random instances, for each $\varepsilon \in \{5, 10, 20\}$.

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				ε		
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				ε		
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/`

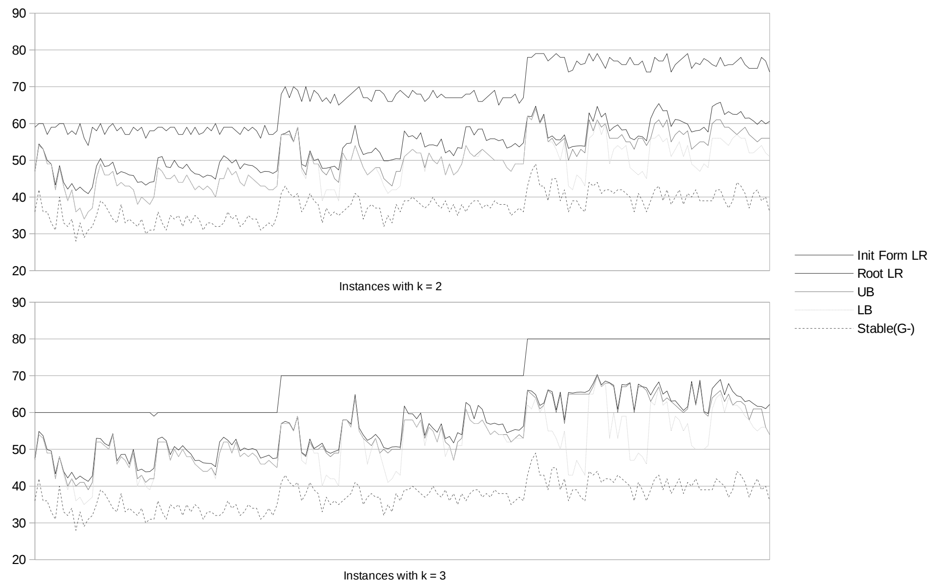

Figure 6: Linear relaxations and bounds obtained by BC on random instances when $k \in \{2, 3\}$.

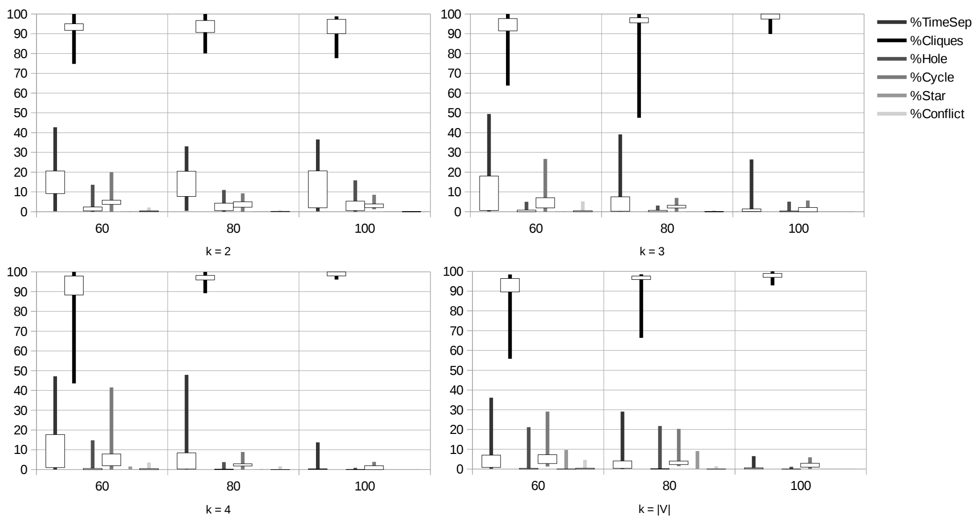

Figure 7: Percentage time and percentage of cuts per family obtained by BC on random instances when $k \in \{2, 3, 4, |V|\}$.

`view_doc.asp?symbol=A/RES/55/56`, December 2000. Accessed on March 2016.

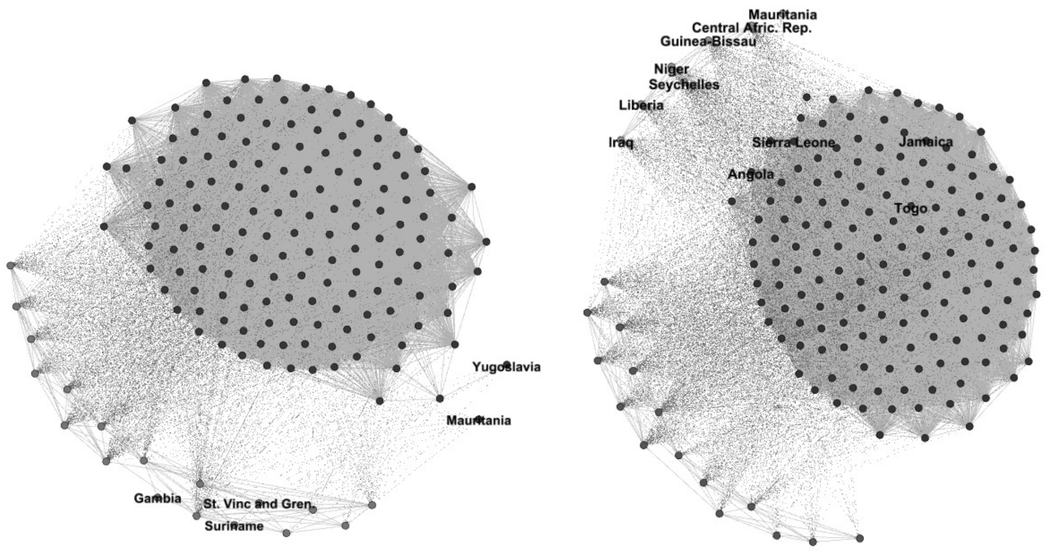

Figure 8: Optimal solutions obtained for instance 3.5 associated with the 55^{th} Session when $k = 2$ (left) and $k = 3$ (right).

[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 43rd 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*, 22:195–209, 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*, 361:172–187, 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*, 35:639–651, 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*, 5:467–498, 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^{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	$Primal_{best}$	Gap %	Time	Nodes	#Solved	$Primal_{best}$	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
	BB+S1S2	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
	BB+S1S2	101.20	0.30	978.04	18.46	10	115.13	0.23	1540.11	49.31	10
n	BB+01	75.11	1.00	1151.70	656.98	3	65.69	1.41	1653.77	370.53	6
	BB+S1S2	73.18	1.18	364.24	41.51	6	60.48	1.76	437.97	94.93	6

Table 6: Branch-and-cut results for instances 3.5

Name	$\alpha(G^-)$	k=2			k=3			k=4			k=n		
		ILS	$Primal_{best}$	(Gap %)/Time	ILS	$Primal_{best}$	(Gap %)/Time	ILS	$Primal_{best}$	(Gap %)/Time	ILS	$Primal_{best}$	(Gap %)/Time
S01	40	46	46	19.53	47	47	17.55	47	47	12.87	47	47	61.63
S02	43	47	48	36.03	48	49	73.58	49	49	31.93	48	49	128.47
S03	49	55	55	47.72	55	55	61.56	55	55	60.55	49	55	131.94
S04	41	43	43	42.42	43	43	74.52	43	43	92.55	43	43	112.60
S05	51	56	56	26.55	56	56	27.73	56	56	37.23	56	56	67.80
S06	43	46	46	75.93	44	47	95.50	47	47	66.18	42	47	152.97
S07	48	52	52	25.86	52	52	35.10	52	52	39.03	34	52	126.92
S08	46	48	48	16.89	48	48	19.55	48	48	19.31	27	48	135.41
S09	50	50	50	84.59	50	50	86.32	50	50	84.16	30	50	147.97
S10	57	60	60	10.06	60	60	14.56	60	60	11.73	60	60	61.41
S11	58	66	66	73.02	66	66	72.80	66	66	73.91	66	66	73.43
S12	61	61	62	253.60	62	62	256.36	62	62	239.34	62	62	206.33
S13	63	64	64	125.33	64	64	135.95	60	64	152.72	61	64	222.33
S14	51	51	51	(0.02)	50	50	(0.04)	45	45	(0.16)	51	51	(0.02)
S15	75	76	76	784.33	70	76	1982.77	76	76	587.40	65	65	(0.18)
S16	64	68	68	(0.12)	66	66	(0.11)	68	68	(0.09)	68	68	(0.09)
S17	72	76	76	(0.05)	77	77	(0.04)	77	77	(0.04)	48	48	(0.67)
S18	92	93	93	1007.61	93	93	1892.31	92	93	3357.31	92	93	2399.80
S20	100	65	106	169.41	71	106	432.85	106	106	534.64	69	106	786.05
S21	91	94	95	2804.21	89	89	(0.09)	69	69	(0.43)	67	67	(0.46)
S22	89	85	85	(0.22)	86	86	(0.23)	86	86	(0.23)	31	31	(2.42)
S23	93	98	98	(0.04)	93	93	(0.16)	99	99	(0.09)	51	51	(1.08)
S24	95	82	82	(0.29)	86	86	(0.30)	102	102	(0.10)	42	42	(1.62)
S25	98	103	103	(0.01)	105	105	(0.06)	105	105	(0.07)	38	38	(1.92)
S26	117	121	121	456.03	122	122	1740.94	123	123	1036.54	31	123	596.96
S27	129	130	130	76.16	130	130	79.10	130	130	126.46	26	130	242.03
S28	120	124	124	535.52	125	125	3284.03	125	125	(0.02)	28	28	(3.46)
S29	124	126	126	332.23	126	126	882.87	126	126	971.99	58	126	1528.74
S30	118	123	123	428.18	123	123	3114.90	123	123	(0.01)	82	82	(0.50)
S31	120	122	122	893.07	122	122	2420.78	122	122	1746.33	7	122	3065.55
S32	131	133	133	251.62	134	134	314.30	134	134	331.81	132	134	485.86
S33	135	40	139	213.68	139	139	545.66	139	139	455.68	63	139	644.23
S34	133	135	135	418.84	135	135	950.92	135	135	789.71	79	135	1247.82
S35	131	132	132	934.63	132	132	(0.02)	132	132	(0.01)	32	32	(3.16)
S36	142	146	146	126.01	129	146	508.52	146	146	358.68	50	146	582.12
S37	136	138	138	812.95	138	138	1462.41	121	138	1903.58	80	138	1460.32
S38	136	141	141	564.52	142	142	670.84	142	142	1237.79	109	142	1760.66
S39	136	141	141	180.12	127	141	591.24	127	141	556.62	98	141	617.65
S40	127	115	128	3188.80	128	128	2839.25	117	128	3060.66	61	128	3186.73
S41	149	152	152	156.23	152	152	293.31	152	152	273.88	131	152	618.80
S42	156	157	157	60.81	157	157	129.25	157	157	117.20	138	157	376.22
S43	149	150	150	351.38	133	150	1351.98	133	150	1187.99	133	150	1182.57
S44	153	156	156	100.36	156	156	117.77	156	156	127.84	123	156	730.84
S45	149	154	154	99.10	154	154	116.91	154	154	213.87	119	154	1020.79
S46	158	159	159	188.29	159	159	377.01	159	159	724.02	111	159	1219.35
S47	158	169	169	317.52	169	169	1199.07	169	169	1401.89	42	169	2448.80
S48	162	171	176	258.88	173	176	1005.71	173	176	1222.97	70	176	1035.23
S49	160	175	175	147.22	174	175	2801.90	175	175	2294.01	112	175	1234.35
S50	152	167	170	1081.21	167	167	(0.03)	170	170	(0.02)	116	116	(0.49)
S51	166	179	179	123.24	179	179	653.50	179	179	551.21	166	179	1500.57
S52	155	157	157	2798.31	158	158	(0.01)	158	158	(0.03)	49	49	(2.27)
S53	165	167	167	277.15	169	169	438.83	169	169	822.37	89	169	1526.39
S54	159	170	170	230.90	171	171	827.02	171	171	1509.61	39	171	1913.65
S55	159	168	170	1082.71	174	176	1720.09	176	177	1878.74	51	51	(2.47)
S56	151	178	178	236.36	181	181	1406.62	181	181	1109.64	57	181	1815.05
S57	167	180	180	320.46	65	183	1176.36	183	183	1840.63	69	183	2058.14
S58	174	64	184	274.42	187	187	373.89	187	187	273.83	60	187	1056.92
S59	180	184	184	181.74	187	187	293.17	187	187	259.09	54	187	1281.66
S60	171	181	181	247.19	73	182	956.36	182	182	743.23	38	182	2724.05
S61	177	181	181	504.81	182	182	1508.12	182	182	1561.95	22	182	3034.64
S62	179	182	182	520.88	82	183	684.31	183	183	3463.93	52	52	(2.52)
S63	177	185	185	127.82	186	186	349.01	186	186	418.25	180	186	1709.56
Avg.	116.62	116.74	121.35	(0.01)/449.13	114.85	121.74	(0.01)/832.64	120.70	121.80	(0.02)/815.77	67.80	106.90	(0.37)/1059.15

Table 7: Branch-and-cut results for instances 1.8

Name	$\alpha(G^-)$	k=2			k=3			k=4			k=n		
		ILS	$Primal_{best}$	(Gap %)/Time	ILS	$Primal_{best}$	(Gap %)/Time	ILS	$Primal_{best}$	(Gap %)/Time	ILS	$Primal_{best}$	(Gap %)/Time
S01	31	34	34	7.18	35	35	12.65	35	35	10.11	35	35	76.91
S02	34	37	37	3.20	38	38	10.04	38	38	13.40	38	38	65.00
S03	34	38	38	18.78	38	38	12.73	38	38	18.58	38	38	64.51
S04	31	31	31	12.39	31	31	14.79	31	31	17.35	31	31	76.59
S05	36	40	40	16.29	40	40	12.43	40	40	26.54	40	40	72.81
S06	29	30	30	32.91	31	31	40.42	31	31	33.63	30	31	159.89
S07	38	38	38	17.08	38	38	18.10	38	38	17.01	33	38	136.08
S08	38	39	39	8.34	39	39	12.17	39	39	15.19	34	39	134.62
S09	38	38	38	7.59	38	38	11.39	38	38	14.50	38	38	71.26
S10	51	52	52	4.34	52	52	13.22	52	52	13.62	51	52	131.29
S11	55	57	57	73.83	57	57	105.92	57	57	67.32	57	57	106.23
S12	46	47	47	197.25	48	48	165.81	48	48	134.30	48	48	202.09
S13	52	47	52	206.14	50	52	333.19	52	52	264.59	50	52	477.83
S14	40	37	40	254.01	40	40	214.29	36	40	311.30	30	40	352.33
S15	47	40	49	410.55	47	49	363.30	47	49	432.63	49	49	403.33
S16	46	49	49	753.25	49	49	916.64	48	49	1000.57	45	49	771.49
S17	48	46	48	3310.75	48	48	1337.18	46	48	3126.84	43	48	2594.59
S18	76	65	77	504.49	74	77	489.97	66	77	534.74	66	77	957.64
S20	73	69	76	670.67	76	76	538.64	69	76	726.45	69	76	975.57
S21	63	64	64	(0.02)	66	66	(0.03)	66	66	(0.03)	32	32	(1.09)
S22	63	63	63	(0.01)	63	63	(0.12)	63	63	(0.12)	46	46	(0.52)
S23	68	70	70	879.18	70	70	1249.22	70	70	914.81	23	70	1204.54
S24	74	76	76	602.79	78	78	644.69	78	78	706.18	78	78	843.99
S25	65	66	66	(0.09)	67	67	(0.10)	68	68	(0.10)	63	63	(0.17)
S26	89	91	91	1183.03	92	92	1193.68	92	92	1347.25	10	92	1770.19
S27	113	114	114	67.94	114	114	166.24	114	114	168.20	114	114	240.30
S28	114	117	117	261.80	117	117	369.00	117	117	370.38	65	117	628.31
S29	112	114	114	421.88	114	114	437.76	114	114	565.35	112	114	995.74
S30	105	109	109	756.30	107	109	1424.75	109	109	971.31	19	109	1456.96
S31	115	117	117	244.81	117	117	336.76	117	117	455.42	115	117	933.29
S32	125	127	127	88.45	127	127	284.06	127	127	170.39	127	127	143.56
S33	112	115	115	1699.44	115	115	1203.56	115	115	1460.63	112	115	1762.92
S34	117	117	117	1390.33	117	117	1534.38	117	117	1269.14	63	117	2156.17
S35	116	117	117	1033.33	117	117	1130.24	117	117	942.27	71	117	1540.13
S36	122	124	124	641.88	124	124	1124.11	124	124	1393.92	124	124	803.52
S37	121	122	122	802.43	122	122	1185.10	122	122	964.55	68	122	1606.58
S38	123	125	125	989.66	125	125	1659.06	125	125	1581.64	125	125	1693.82
S39	126	129	129	310.00	129	129	979.17	129	129	636.44	96	129	1193.85
S40	120	120	120	637.04	120	120	535.27	120	120	822.33	115	120	1417.15
S41	143	144	144	361.26	144	144	281.54	144	144	202.87	144	144	400.65
S42	146	147	147	167.68	147	147	182.38	147	147	188.07	147	147	376.45
S43	142	144	144	382.48	144	144	382.44	144	144	514.97	144	144	325.67
S44	143	144	144	357.76	144	144	636.19	144	144	443.30	138	144	1149.25
S45	141	146	146	269.73	146	146	307.03	146	146	415.06	142	146	792.55
S46	137	138	138	2026.44	138	138	2914.64	138	138	2476.26	122	138	3102.09
S47	137	146	146	1376.79	146	146	2080.97	146	146	1940.18	37	146	2745.67
S48	148	158	158	404.40	158	158	600.37	158	158	775.90	95	158	1546.58
S49	150	163	164	547.51	163	164	1232.74	163	164	1596.74	45	164	2026.43
S50	138	154	154	527.80	154	154	1336.27	154	154	1290.59	154	154	986.10
S51	150	161	161	1004.81	161	161	2027.71	161	161	1751.55	73	161	3185.62
S52	139	142	142	1321.96	142	142	1551.90	142	142	1236.85	105	142	2620.97
S53	149	151	151	1176.38	151	151	1414.39	151	151	1498.61	115	151	2129.75
S54	144	143	154	679.77	154	154	1182.66	154	154	1187.51	16	154	1959.83
S55	149	157	157	693.50	163	163	602.38	163	163	1301.74	88	163	2511.84
S56	137	162	162	522.46	162	162	1277.85	162	162	1096.91	56	162	3193.99
S57	147	157	157	2140.23	155	158	3130.77	158	158	2456.40	117	158	3382.21
S58	157	132	165	1101.29	165	165	862.89	131	165	1559.18	131	165	2536.97
S59	158	160	160	2248.27	160	160	2101.60	133	160	2235.95	41	160	3232.25
S60	155	163	163	1746.40	163	163	1831.65	163	163	2411.39	155	163	2474.71
S61	162	165	165	1143.07	166	166	970.84	166	166	1422.29	105	166	2474.71
S62	150	153	153	(0.01)	154	154	(0.01)	153	153	(0.01)	69	69	(1.24)
S63	165	172	172	269.71	173	173	660.92	173	173	415.27	171	173	1879.87
Avg.	101.5	103.75	105.09	(0.00)/672.22	105.20	105.41	(0.00)/821.55	103.98	105.41	(0.00)/860.97	77.62	103.16	(0.04)/1235.12

Table 8: Comparison of results obtained on UNGA instances by each solution method.

n	k=2					k=3					k=4					k=n				
	BB			BC		BB			BC		BB			BC		BB			BC	
	R_r	#01	#S1S2	R_r^c	#BC	R_r	#01	#S1S2	R_r	#BC	R_r	#01	#S1S2	R_r	#BC	R_r	#01	#S1S2	R_r	#BC
0-100	61.0	12	18	47.5	27	64.3	6	18	47.7	27	65.3	3	19	47.9	27	65.8	9	13	47.7	27
101-150	119.2	0	0	99.6	28	124.3	1	0	111.4	28	126.6	0	0	100.6	26	128.4	0	0	100.1	25
151-200	164.0	2	1	156.0	55	170.3	1	1	156.7	52	172.2	1	1	156.8	52	170.9	0	0	156.4	50
Avg /Sum	114.7	14	19	101.0	110	119.6	8	19	105.3	107	121.3	4	20	101.8	105	121.7	9	13	101.4	102