



A polyhedral study of the maximum edge subgraph problem

Flavia Bonomo^{a,*}, Javier Marengo^b, Daniela Saban^{c,b}, Nicolás E. Stier-Moses^d

^a Depto. de Computación, FCEyN, Universidad de Buenos Aires and IMAS-CONICET, Argentina

^b Instituto de Ciencias, Universidad Nacional de General Sarmiento, Argentina

^c Depto. de Computación, FCEyN, Universidad de Buenos Aires, Argentina

^d Graduate School of Business, Columbia University, USA

ARTICLE INFO

Article history:

Received 16 April 2010

Received in revised form 25 July 2011

Accepted 8 October 2011

Available online 4 November 2011

Keywords:

Polyhedral combinatorics

Quasi-cliques

Maximum edge subgraph problem

ABSTRACT

The study of cohesive subgroups is an important aspect of social network analysis. Cohesive subgroups are studied using different relaxations of the notion of clique in a graph. For instance, given a graph and an integer k , the *maximum edge subgraph problem* consists of finding a k -vertex subset such that the number of edges within the subset is maximum. This work proposes a polyhedral approach for this NP-hard problem. We study the polytope associated to an integer programming formulation of the problem, present several families of facet-inducing valid inequalities, and discuss the separation problem associated to these families. Finally, we implement a branch and cut algorithm for this problem. This computational study illustrates the effectiveness of the classes of inequalities presented in this work.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Social network analysis is an important tool to study the relationships and flows between people, organizations, and other entities. Social networks are encoded by graphs with vertices representing the entities, and edges representing interdependencies between them. An important aspect of social network analysis is the detection of *cohesive subgroups*, which are subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties [26]. Although one could potentially define such a subgroup with the concept of clique, this does not provide a full picture because in practice missing edges frequently exist but that is not a strong enough reason to claim that two vertices cannot belong to the same group. For example, if two individuals in a social network are not friends because they fought but share most of their friends, they would belong to the same social group. For this reason, it is more satisfactory to define a cohesive subgroup using a relaxation of the definition of clique.

Let $G = (V, E)$ be the graph that represents the social network of interest. One possible approach to study cohesive subgroups is the use of *quasi-cliques*, which are subgraphs with a pre-specified minimum edge density. Here, the density of a subgraph is the quotient between the number of edges with both endpoints in the subgroup and the total number of edges. Formally, given a real number $0 \leq \gamma \leq 1$, a subgraph $G' = (N', E') \subseteq G$ is a γ -quasi-clique if $2|E'|/(|N'|(|N'| - 1)) \geq \gamma$. In the example of the social network of friends referred to earlier, missing relationships would represent that pairs of individuals in the group may not be friends with each other. In earlier work, we strongly relied on quasi-cliques to study the network of bilateral investment treaties signed between countries of the world [24]. Besides having used them to identify cohesive subgroups, they also allowed us to compare different random topologies. Indeed, cliques are inexact because there

* Corresponding author.

E-mail addresses: fbonomo@dc.uba.ar (F. Bonomo), jmarengo@ungs.edu.ar (J. Marengo), dsaban@dc.uba.ar (D. Saban), stier@gsb.columbia.edu (N.E. Stier-Moses).

is no guarantee that randomly generated edges will form the same cliques as in the original graph. Quasi-cliques, instead, can capture the topological structure regardless of which particular random edges are present in the network.

Since quasi-cliques are characterized by their size and density, they have been studied by considering two formulations that are dual of each other, much like other bi-criteria optimization problems. One can either be given (i) a specified edge density $\gamma \in [0, 1]$, and find the largest γ -quasi-clique, or be given (ii) a size k of a subgraph, and find the densest set of k vertices. The second approach – the one we follow in this paper – is known in the graph and optimization literature as the *maximum edge subgraph problem* (MESP) [4] or *dense/densest/heaviest k -subgraph problem* [12] or *k -cluster problem* [8]. Concretely, given an integer $k < |V|$, the MESP consists of finding a vertex subset $A \subset V$ with $|A| = k$ such that $|E(A)|$ is maximum, where $E(A) = \{ij \in E : \{i, j\} \subset A\}$.

The maximum clique problem reduces to the MESP, hence the latter is NP-hard [3]. Indeed, if the MESP can be solved in polynomial time, then we can polynomially search the greatest k for which the answer to the MESP is a complete subgraph, thus solving the maximum clique problem. This problem remains NP-hard even for some simpler classes of graphs, such as comparability, triangle-free and chordal graphs [8]. The complexity of the problem motivated [1,4,12,15,23] to look for approximation algorithms and heuristics, [8,19,20] to consider restricted classes of graphs, and [6] to introduce several integer programming formulations. In this work, we consider the polytope associated to the formulation *MIP1* of [6]. We introduce several families of facet-inducing valid inequalities and discuss the separation problem associated to these families. Note that [21,22], who independently considered a generalization of our problem, also present families of inequalities that are facet-inducing. Some of those facets are related to ours, as we point out when we present our facets. Although the main contribution of our paper lies with the study of the polytope and its facets, for completeness we also perform a small computational study to test the empirical strength of the families of facets and comment on which are most effective computationally.

The remainder of this paper is organized as follows. Section 2 presents the formulation and studies some of its properties. In Section 3, we introduce some classes of valid inequalities for this formulation, discuss in what situations they induce facets of the associated polytope, and explore the computational complexity of the corresponding separation problems. Section 4 describes the implementation of a branch and cut algorithm based on these classes of valid inequalities and reports on our computational results. Finally, we conclude with some remarks and directions for future research.

2. Integer programming formulation

To represent quasi-cliques, we introduce a binary *vertex variable* x_i for every $i \in V$. We set $x_i = 1$ if and only if the vertex i belongs to the k -subset $A \subset V$, which defines a feasible solution. For every $ij \in E$, we introduce a binary *edge variable* z_{ij} that satisfies that $ij \in E(A)$ when $z_{ij} = 1$. Notice that z_{ij} and z_{ji} denote the same variable because edges are undirected. With these definitions, the maximum edge subgraph problem can be formulated using the following integer program. To be consistent with the literature, we refer to this formulation as *MIP1*.

$$\max \sum_{ij \in E} z_{ij} \quad (1a)$$

$$\text{s.t. } \sum_{i \in V} x_i = k \quad (1b)$$

$$z_{ij} \leq x_i \quad \forall ij \in E \quad (1c)$$

$$z_{ij} \leq x_j \quad \forall ij \in E \quad (1d)$$

$$x_i \in \{0, 1\} \quad \forall i \in V \quad (1e)$$

$$z_{ij} \in \{0, 1\} \quad \forall ij \in E. \quad (1f)$$

Here, the objective counts the number of edges in the subgraph, the first constraint guarantees that the subgraph has the desired size, and the second and third constraints imply that only edges with both endpoints in the subset can be selected. Note that (1f) can be dropped from the formulation, thus letting the edge variables be continuous and free variables. In an optimal solution, the edge variables will be binary automatically.

We define $P(G, k) \subseteq \mathbb{R}^{|V|+|E|}$ to be the convex hull of the vectors (x, z) satisfying constraints (1b)–(1f). Moreover, we let $P_{LP}(G, k) \subseteq \mathbb{R}^{|V|+|E|}$ be the linear relaxation of $P(G, k)$, as given by constraints (1b)–(1d), $0 \leq x_i \leq 1$ for $i \in V$, and $0 \leq z_{ij} \leq 1$ for $ij \in E$. The next result implies that the polytope does not lose more dimensions than the single one lost by the equality constraint (1b). This proof uses standard arguments and is therefore omitted.

Theorem 1. *The dimension of the polytope $P(G, k)$ is $|V| + |E| - 1$.*

3. Valid inequalities

Each of the following subsections introduces a class of facet-inducing valid inequalities for $P(G, k)$ and studies the complexity of the associated separation problem. All these classes arise from combinatorial structures in the original graph

such as trees and matchings. These inequalities were designed by analyzing the facets of simple instances that allow a complete characterization of the polytope and generalizing them to arbitrary instances using ideas from other problems such as those studied in [9,25].

3.1. Neighborhood inequalities

We start with our simplest class of facet-defining inequalities, defined by a specific vertex and a subset of vertices. We refer to the *neighborhood* of a vertex $i \in V$ by $N(i) := \{j \in V : ij \in E\}$.

Definition 1. For a vertex $i \in V$ and a set $A \subseteq V \setminus \{i\}$ with $|A| \leq k - 2$, a *generalized neighborhood inequality* associated with the vertex i and the set A is defined by

$$\sum_{j \in A} x_j + \sum_{j \in N(i) \setminus A} z_{ij} \leq |A| + (k - |A| - 1)x_i. \tag{2}$$

A particularly simple case is given by a vertex i and the set $A = \emptyset$, from where we have that $\sum_{j \in N(i)} z_{ij} \leq (k - 1)x_i$. We refer to this inequality as the *neighborhood inequality* associated with vertex i . The neighborhood inequalities are equivalent to the *star subgraphs* inequalities introduced by Sherali and Cole Smith [25] and to the *star inequalities* defined by Mehrotra in [21]. The generalized neighborhood inequalities are a facet-inducing superclass of these inequalities.

The next result establishes conditions under which generalized neighborhood inequalities induce facets of $P(G, k)$. They require that either i is well connected or A is big enough. The case of $k = |V| - 1$ is excluded but that is not important because the optimal solution to the MESP is easy to compute in that case.

Theorem 2. *The generalized neighborhood inequality (2) is valid for $P(G, k)$. Moreover, if $|N(i) \cup A| \geq k$ and $k \leq |V| - 2$, then (2) is facet-inducing for $P(G, k)$.*

Proof. Let $(x, z) \in P(G, k) \cap \mathbb{Z}^{|V|+|E|}$ be a feasible solution. If $x_i = 0$, then $z_{ij} = 0$ for every $j \in N(i)$, and (2) is satisfied by (x, z) because $\sum_{j \in A} x_j \leq |A|$. If $x_i = 1$, there remain at most $k - 1$ variables set to 1, which implies that

$$\sum_{j \in A} x_j + \sum_{j \in N(i) \setminus A} z_{ij} \leq \sum_{j \in A} x_j + \sum_{j \in N(i) \setminus A} x_j \leq k - 1 = |A| + (k - |A| - 1)x_i.$$

In both cases, the solution (x, z) satisfies (2). Since (x, z) is an arbitrary integer point in $P(G, k)$, then (2) is valid for this polytope.

Let F be the face of $P(G, k)$ defined by (2) and suppose $\lambda^T(x, z) = \lambda_0$ for every $(x, z) \in F$. We shall prove that (λ, λ_0) is a linear combination of the coefficient vector of the generalized neighborhood inequality (2) and the coefficient vector of the model constraint (1b), thus showing that F is a facet of $P(G, k)$.

Claim 1. $\lambda_{z_{jl}} = 0 \forall j, l \in E : j, l \neq i$. Consider the feasible solution $w^1 = (x, 0) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$, $w_{x_j}^1 = 1$, $w_{x_l}^1 = 1$, $w_{x_i}^1 = 0$, and the number of vertices $v \in V$ such that $w_{x_v}^1 = 1$ is exactly k . Notice that j and l may be vertices in A . This construction is feasible since $|V| \geq k + 2$ and $|A| \leq k - 2$. Since $jl \in E$, let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{jl}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{z_{jl}} = 0$.

Claim 2. $\lambda_{x_j} = \lambda_{x_l} \forall j, l \in V \setminus A : j, l \neq i$. Consider the feasible solution $w^1 = (x, 0) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$, $w_{x_j}^1 = 1$, $w_{x_l}^1 = 1$, $w_{x_i}^1 = 0$, and the number of vertices $v \in V$ such that $w_{x_v}^1 = 1$ is exactly k . This construction is feasible since $|V| \geq k + 2$ and $|A| \leq k - 2$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_l}^2 = 1$, $w_{x_j}^2 = 0$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{x_j} = \lambda_{x_l}$.

Claim 3. $\lambda_{z_{ij}} = \lambda_{z_{il}} \forall j, l \in N(i) \setminus A$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$, $w_{x_i}^1 = 1$, $w_{x_l}^1 = 0$, $w_{x_t}^1 = w_{z_{it}}^1 = 1$ for a subset of exactly $(k - |A| - 1)$ vertices in $N(i) \setminus (A \cup \{l\})$ including the vertex j , and $w_{z_{ij}}^1 = 1$. It is easy to verify that this construction is feasible, since $|N(i) \setminus A| \geq k - |A|$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{il}}^2 = 1$, $w_{x_j}^2 = 0$, $w_{z_{ij}}^2 = 0$ and $w_{z_{il}}^2 = 1$. Since $\lambda^T w^1 = \lambda^T w^2$, we obtain $\lambda_{z_{ij}} + \lambda_{x_j} = \lambda_{z_{il}} + \lambda_{x_l}$. As j, l are vertices in $V \setminus (A \cup \{i\})$, by Claim 2 we know $\lambda_{x_j} = \lambda_{x_l}$, and thus we conclude $\lambda_{z_{ij}} = \lambda_{z_{il}}$.

Claim 4. $\lambda_{z_{ia}} = 0 \forall a \in A, ia \in E$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$, $w_{x_i}^1 = 1$, $w_{x_j}^1 = 1$ for a subset of exactly $(k - |A| - 1)$ vertices $j \in N(i) \setminus A$, and $w_{z_{ij}}^1 = 1$ iff $j \in N(i) \setminus A$ and $w_{x_j}^1 = 1$. It can be seen that this construction is feasible, since $|N(i) \setminus A| \geq k - |A|$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{ia}}^2 = 1$. Since a is an arbitrary vertex of A , we conclude $\lambda_{z_{ia}} = 0$.

Claim 5. $\lambda_{x_a} = \lambda_{z_{ij}} + \lambda_{x_j} \forall a \in A, j \in N(i) \setminus A$. Consider the feasible solution w^1 as in Claim 4, constructed in such a way that $w_{x_j}^1 = 0$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_a}^2 = 0, w_{x_j}^2 = 1$ and $w_{z_{ij}}^2 = 1$. Again, these constructions are feasible since $|N(i) \setminus A| \geq k - |A|$. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{x_a} = \lambda_{z_{ij}} + \lambda_{x_j}$.

Claim 6. $\lambda_{x_l} = \lambda_{x_i} + (k - |A| - 1)\lambda_{z_{ij}} \forall j \in N(i) \setminus A, l \in V \setminus (A \cup \{i\})$. Consider the feasible solution $w^1 = (x, 0) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$ and $w_{x_t}^1 = 1$ for a subset of exactly $(k - |A|)$ vertices $t \in N(i) \setminus A$ including the vertices j and l (all the remaining x -variables are set to zero). It can be seen that this construction is feasible, since $|N(i) \setminus A| \geq k - |A|$ and $|A| \leq k - 2$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_l}^2 = 0, w_{x_i}^2 = 1$ and $w_{z_{it}}^2 = 1$ for the remaining $(k - |A| - 1)$ vertices $t \in N(i) \setminus A$ with $w_{x_t}^1 = 1$, and keeping the remaining variables unchanged. Since $l \in N(i) \setminus A$ and Claim 3 implies $\lambda_{z_{ij}} = \lambda_{z_{il}}$ for every $j, l \in N(i) \setminus A$, we conclude $\lambda_{x_l} = \lambda_{x_i} + (k - |A| - 1)\lambda_{z_{ij}}$.

We let $\alpha := \lambda_{z_{ij}}$ for $j \in N(i) \setminus A$, which does not depend on the choice of j because of Claim 3. We define $\beta := \lambda_{x_r}$ for $r \in V \setminus (A \cup \{i\})$, which is well-defined by Claim 2. From here, $\lambda_{x_i} = \beta - (k - |A| - 1)\alpha$ by Claims 2 and 6, $\lambda_{x_a} = \alpha + \beta$ for every $a \in A$ by Claim 5, and $\lambda_{z_{jl}} = 0$ for every $jl \notin \{ij : j \in N(i) \setminus A\}$ by Claims 1 and 4. We conclude that λ is a linear combination of the coefficient vector π of the generalized neighborhood inequality (2) and the coefficient vector π' of the model constraint (1b); i.e., $\lambda = \alpha\pi + \beta\pi'$. Let π_0 and π'_0 be the independent terms of (2) and (1b), respectively. For every $(x, z) \in F$ we have

$$\lambda_0 = \lambda^T(x, z) = (\alpha\pi + \beta\pi')^T(x, z) = \alpha\pi^T(x, z) + \beta\pi'^T(x, z) = \alpha\pi_0 + \beta\pi'_0, \tag{3}$$

hence (λ, λ_0) is a linear combination of (π, π_0) and (π', π'_0) , and (2) induces a facet of $P(G, k)$. \square

Note that if G is a star, then A can have $k - 1$ vertices (i.e., $|A| \leq k - 1$) and still be facet defining.

We now study the separation problem associated with the generalized neighborhood inequalities. Considering an input vector $(x, z) \in P_{LP}(G, k)$, this problem consists of determining whether the point (x, z) is separated by a generalized neighborhood inequality or not. To prove that this problem is solvable in polynomial time, we need the following lemma. For $i \in V$, we define $\bar{N}(i) := V \setminus (\{i\} \cup N(i))$ as the complement of the neighborhood of i .

Lemma 1. *If the generalized neighborhood inequality associated with the vertex i and the set A is violated by the point $(x, z) \in P_{LP}(G, k)$, then the corresponding inequality associated with the vertex i and the set*

$$A' = A \setminus (\{j \in A \cap N(i) : x_i + x_j \leq 1 + z_{ij}\} \cup \{j \in A \cap \bar{N}(i) : x_i + x_j \leq 1\})$$

is also violated.

Proof. We show that, when sequentially removing the vertices in $\{j \in A \cap N(i) : x_i + x_j \leq 1 + z_{ij}\} \cup \{j \in A \cap \bar{N}(i) : x_i + x_j \leq 1\}$ from A , the difference between the RHS and the LHS of the inequality never increases. Let $j \in A$ and let $A' = A \setminus \{j\}$. As the generalized neighborhood inequality associated with the vertex i and the set A is violated, we know that

$$\sum_{v \in A'} x_v + x_j + \sum_{v \in N(i) \setminus A} z_{iv} > |A| + (k - |A| - 1)x_i. \tag{4}$$

If $j \in N(i)$, the generalized neighborhood inequality associated with i and A' is violated if

$$\sum_{v \in A'} x_v + z_{ij} + \sum_{v \in N(i) \setminus A} z_{iv} > |A'| + (k - |A'| - 1)x_i. \tag{5}$$

The difference between the LHS and the RHS of (5) is at least the difference between the LHS and the RHS of (4) if and only if $x_i + x_j \leq 1 + z_{ij}$. This shows that we can safely remove from A the vertex $j \in N(i)$ if $x_i + x_j \leq 1 + z_{ij}$.

If $j \notin N(i)$, the generalized neighborhood inequality associated with i and A' is violated if

$$\sum_{v \in A'} x_v + \sum_{v \in N(i) \setminus A} z_{iv} > |A'| + (k - |A'| - 1)x_i. \tag{6}$$

In this case, the difference between the LHS and the RHS of (6) does not decrease with respect to (4) if and only if $1 - x_i \geq x_j$, hence we can safely delete j from A if this property holds. As the vertex $j \in A$ is an arbitrary vertex, the lemma follows. \square

We are now in a position to address the separation of generalized neighborhood inequalities. If $x_i = 0$, no such inequality associated with the vertex i will be violated because $\sum_{j \in A} x_j \leq |A|$. When $x_i > 0$, Lemma 1 implies that we only need to check the generalized neighborhood inequalities associated with the sets A described in the lemma. The following result shows that Algorithm 1, a straightforward implementation of this idea, can separate this class in polynomial time.

Theorem 3. *The generalized neighborhood inequalities can be separated in $O(n^2 \log n)$ time using Algorithm 1.*

Notice that the bottleneck of the algorithm, which defines the running time, is sorting the vertices with respect to ω_j for every vertex i with $x_i > 0$.

Algorithm 1 Separation of the generalized neighborhood inequalities (GNI)

Input: A point $(x, z) \in P_{LP}(G, k)$.
Output: A violated GNI, or a certificate that no such inequality exists.
for $i \in V$ such that $x_i > 0$ **do**
 $A = \emptyset$
 for $j \in N(i)$ **do**
 $\omega_j = x_j - z_{ij}$
 end for
 for $j \in V \setminus (N(i) \cup \{i\})$ **do**
 $\omega_j = x_j$
 end for
 for $j \in V \setminus \{i\}$ in decreasing order of ω_j **do**
 if $|A| < k - 2$ and $\omega_j > 1 - x_i$ **then**
 Add j to A
 end if
 end for
 Check if the GNI associated with the vertex i and the set A is violated
end for

3.2. Matching inequalities

We now introduce another class of inequalities that use a matching (i.e., a maximal set of pairwise disjoint edges) instead of a single vertex i .

Definition 2. Let $A \subseteq V$ and let B be a maximal matching of $E(V \setminus A)$. A *matching inequality* associated with the set A and the matching B is defined by

$$\sum_{i \in A} x_i + \sum_{ij \in B} z_{ij} \leq \frac{|A| + k - 1}{2}. \tag{7}$$

Matching inequalities are related to the *odd-set inequalities* introduced by Edmonds [9]. For our problem, though, we consider edges instead of vertices, and we take into account the parity of k and allow the set A to be arbitrary.

The next results prove that matching inequalities are valid as long as $|A|$ and k do not have the same parity, and that they are facet-defining when the cardinalities of A and B have the correct values. Similarly to before, the case of $k > |V| - 3$ can be solved in polynomial time so we can disregard that case. We will just prove that the inequality is valid; the proof that it is facet-defining is similar to that of [Theorem 2](#) and is deferred to the [Appendix](#).

Theorem 4. *The matching inequality (7) is valid for $P(G, k)$ if $|A| + k$ is odd. In addition, if $k - 2|B| + 1 \leq |A| \leq k - 1$ and $k \leq |V| - 3$, then (7) is facet-defining.*

Proof. Let $(x, z) \in P(G, k) \cap \mathbb{Z}^{|V|+|E|}$ be an arbitrary feasible solution, and call $r = \sum_{i \in A} x_i$. Since B is a matching, we have $\sum_{ij \in B} z_{ij} \leq (k - r)/2$, hence

$$\sum_{i \in A} x_i + \sum_{ij \in B} z_{ij} \leq r + (k - r)/2 = (r + k)/2. \tag{8}$$

If $r = |A|$, since $|A| + k$ is odd, the RHS of (8) can be replaced by $(|A| + k - 1)/2$, and the resulting inequality coincides with (7). If $r \leq |A| - 1$ then (8) dominates (7), hence the latter is a valid inequality for $P(G, k)$. \square

We now address the separation problem for this family of inequalities. Recall that the maximum weight matching problem can be solved in $O(|V|^3)$ time [10]. We define $V(B) \subseteq V$ to be the set of endpoints of the edges in B .

Theorem 5. *The matching inequalities can be separated in $O(|V|^3)$ time if k is odd, and can be separated in $O(|V|^4)$ time if k is even.*

Proof. Let us first consider the case when $|A|$ must be even because k is odd. We consider the complete graph $G' = (V, E')$ with edge weights $w_{ij} = \max\{z_{ij}, x_i + x_j - 1\}$ for $ij \in E$ and $w_{ij} = x_i + x_j - 1$ for $ij \notin E$. Let $M \subseteq E'$ be a maximum-weight matching in G' , and define $E_A = \{ij \in M : w_{ij} = x_i + x_j - 1\}$. We have $w(M) > (k - 1)/2$ if and only if there exists some violated matching inequality. For the forward direction, it suffices to consider the inequality associated with $A = V(E_A)$ and $B = M \setminus E_A$. Note that, as $z_{ij} \geq 0$, we can always extend B to a maximal matching of $G \setminus A$. For the converse, add to the matching B an arbitrary perfect matching of the complete graph induced by A . Finally, in the case when $|A|$ must be odd, we repeat the previous procedure $|V|$ times by fixing each vertex i in A and by considering $G' \setminus \{i\}$. In this case, there exists some violated matching inequality if and only if $w(M) > k/2 - x_i$. \square

3.3. Forest inequalities

We now consider a family of inequalities defined over a subset of vertices and over a forest inside that subset.

Definition 3. Let $A \subseteq V$ be a vertex subset with cardinality $|V| - k + 1$, and let $T \subseteq E(A)$ be an acyclic edge subset of A . A forest inequality associated with A and T is defined by

$$1 + \sum_{ij \in T} z_{ij} \leq \sum_{i \in A} x_i. \tag{9}$$

The theorem below states that this family of inequalities is always valid and it is facet-defining when paths in the forest T are not too long. The proof that the family is facet-defining is not included because it is similar to [Theorem 2](#).

Theorem 6. The forest inequality (9) is valid for $P(G, k)$. In addition, if for every $ij \in E(A) \setminus T$ there exists a path in T from i to j of length at most $k - 1$, then (9) is facet-defining.

Proof. Let $(x, z) \in P(G, k) \cap \mathbb{Z}^{|V|+|E|}$ be a feasible solution. Letting $M \subseteq T$ be the subset of edges ij in the forest such that $z_{ij} = 1$, the LHS of (9) equals $1 + |M|$. This value is upper bounded by $|V(M)|$ because M is acyclic. If M is empty then, as $|V \setminus A| = k - 1$, there must be at least one vertex $i \in A$ such that $x_i = 1$, so the inequality holds. If M is nonempty, the inequality also holds because $x_i = 1$ for every $i \in V(M)$. Since (x, z) is an arbitrary integer point in $P(G, k)$, then (9) is valid for this polytope. \square

The separation problem for forest inequalities, defined as follows, turns out to be NP-complete.

FOREST INEQUALITIES SEPARATION

Instance: A graph G , an integer k and a point $w = (x, z) \in P_{LP}(G, k)$

Question: Does w violate some forest inequality?

Theorem 7. FOREST INEQUALITIES SEPARATION is NP-complete.

Proof. We can easily check that this problem belongs to the class NP, since we can nondeterministically generate a vertex subset $A \subset V$ and a forest $T \subseteq E(A)$, and verify in deterministic polynomial time whether the forest inequality associated with A and T is violated by the point w or not. To complete the proof, we construct a polynomial reduction from the edge-weighted version of the r -CARDTREE problem [13] to FOREST INEQUALITIES SEPARATION. An instance of r -CARDTREE is given by a tuple (H, r, p) , where $H = (V_H, E_H)$ is an edge-weighted graph with weight function $w_H : E_H \rightarrow \mathbb{R}$, $r \in \mathbb{N}$ is a nonnegative integer such that $r \leq |V_H|$, and $p \in \mathbb{R}$. An instance is affirmative when H contains a tree of exactly r edges whose weight is less than p , and negative otherwise. This problem is known to be (strongly) NP-hard [13].

The edge weights can be assumed to be positive (if some of them are negative, let $\alpha = -\min\{w_H(ij) : ij \in E_H\}$, add α to all the edge weights, and modify the value of p by adding $r\alpha$). We can also trivially assume $r > 2$ and $w_H(ij) \leq p$ for every edge $ij \in E_H$. Finally, notice that without loss of generality we can always assume $|V_H| > cr$, where $c \geq 1$ is a positive number, by adding a suitable number of isolated vertices.

To reduce this instance to an instance of FOREST INEQUALITIES SEPARATION, we keep the same graph structure and construct a feasible solution w for which the two problems will be equivalent. The goal is getting a violated forest inequality from w when there exists a tree of cardinality r with weight less than p . We can safely assume that $|V_H| \geq 2r + 4$ (otherwise, we add the necessary isolated vertices).

- Set the structure of the graph $G(V, E)$ as the original one $H(V_H, E_H)$. Set $k = n - r$, where $n = |V_H|$.
- We construct a feasible solution $w = (x, z) \in \mathbb{R}^{|V|+|E|}$. To have all vertices with the same value, we set $x_i = 1 - r/n$ in each vertex $i \in V$. From here, the RHS of (9) equals $(r + 1)(n - r)/n$.
- To get trees that violate (9) to have small cardinality, we need to make z inversely correlated with weights w_H . We choose the functional form $z_{ij} = 1 - \rho(w_H(ij))r/n$ for a carefully constructed function $\rho(w) = 1 + (w/p + 1)/(r + 1) \geq 1$.

The function $\rho(w)$ above was specifically chosen to get the following important property, which follows after some algebra:

$$\sum_{ij \in T} w_H(ij) < p \Leftrightarrow 1 + \sum_{ij \in T} z_{ij} > \sum_{i \in V(T)} x_i. \tag{10}$$

This construction is polynomial in the size of H . We now verify that $w \in P_{LP}(G, n - r)$ by checking that w satisfies all the constraints of the relaxed polytope. Constraint (1b) is satisfied by construction, while (1c) and (1d) hold because $\rho(w) \geq 1$. Finally, the relaxed versions of (1e) and (1f) are verified because $n \geq 2r + 4$ and $w_H(ij) \leq p$. Putting all together, $w \in P_{LP}(H, n - r)$. To complete the proof, we show that the prescribed transformation maps affirmative instances of r -CARDTREE onto affirmative instances of FOREST INEQUALITIES SEPARATION and conversely. In other words, we need to prove that (H, r, p) is affirmative, i.e., there exists a tree T of $r + 1$ vertices whose edge weight is at most p , if and only if the point w as constructed above violates a forest inequality.

Algorithm 2 Heuristic for the separation of forest inequalities

Input: A point $(x, z) \in P_{LP}(G, k)$.
Output: A set of violated forest inequalities (which may be empty)
Construct the directed graph $G_H = (V_H, E_H)$ in the following way:
 $V_H = \{i \in V : 0 < x_i < 1\}$,
 $E_H = \{i \rightarrow j : ij \in E, z_{ij} = x_j > 0\}$.
Let $R = \{i \in V : x_i = 0\}$ and $r = |R|$.
Set $S = \emptyset$.
for $i \in V_H$ **do**
 Using BFS build a tree T starting from vertex i until the tree has $|V| - k + 1$ vertices, or until no more vertices can be added.
 if T has at least $|V| - r - k + 1$ vertices **then**
 The inequality associated with the edge set T and the vertex set $A = V(T) \cup U$, where $U \subseteq R$ and $|U| = |V| - k + 1 - |V(T)|$, is violated by $1 - x_i$.
 Add the inequality to S .
 end if
end for
Return S .

(\Rightarrow) By (10), we conclude that the forest inequality associated with $A = V(T)$ and the tree T is violated.
(\Leftarrow) Conversely, suppose that the forest inequality associated with the vertex set A and the forest F is violated. Set A is such that $|A| = r + 1$ and $V(F) \subseteq A$. If F is a tree spanning A , the claim follows. Otherwise, $|E(F)| \leq r - 1$. Since $w_G(ij) \leq (n - r)/n$ for every edge ij of G , and $n > 2r$, then $1 + \sum_{ij \in E(F)} z_{ij} \leq 1 + (r - 1)(n - r)/n \leq (r + 1)(n - r)/n = \sum_{i \in V(F)} x_i$, a contradiction. Therefore, the transformation maps affirmative instances of r -CARDTREE onto affirmative instances of forest inequalities separation and conversely, hence the latter is NP-complete. \square

Since the separation problem for forest inequalities is NP-complete, we propose a heuristic procedure for tackling this problem. Our heuristic, which is shown in Algorithm 2, decides which vertices to place in A by attempting to find trees that violate a forest inequality. We let $V_H := \{i \in V : 0 < x_i < 1\}$ and $R := \{i \in V : x_i = 0\}$. First we construct a directed graph G_H on the vertices of V_H , by adding a directed edge from vertex i to vertex j , only if the value of the variable z_{ij} equals the value of x_j . The algorithm tries to find a maximum cardinality spanning tree rooted at a vertex $i \in V_H$. The construction of G_H guarantees that each vertex, with the exception of the root, has the same value as the directed edge entering it. If the cardinality of the tree is $|V| - k + 1$, then we have found a violated inequality. If not, we can try to expand A by adding vertices in R . Note that this will not affect the values at either side of the inequality, that will end up violated by $1 - x_i$. We refer the reader to Section 4 for a discussion on the effectiveness of this heuristic in finding violated inequalities.

3.4. Tree inequalities

In this section we consider a family of inequalities that depends on a subset of vertices as before, but now we take a tree induced by the vertices outside the subset. For a tree T , we define $L_T \subset V_T$ to be the set of leaves of T and $\bar{L}_T := V_T \setminus L_T$ to be the set of interior vertices; i.e., \bar{L}_T is the set containing all the vertices $i \in V_T$ such that $d_T(i) > 1$. We denote the number of leaves of T by $\ell := |L_T|$.

In the sequel, we will use the following notation. For a subgraph $H \subseteq G$ and a vertex $i \in V_H$, denote by $N_H(i) := \{j : ij \in E_H\}$ the set of neighbors of i in H , and by $d_H(i) := |N_H(i)|$ the degree of i in H .

Definition 4. Let $A \subset V$ be a (possibly empty) vertex subset with $|A| < k$. In addition, let $T = (V_T, E_T) \subseteq G \setminus A$ be a tree on $k - |A| + 1$ vertices. A tree inequality associated with the subset A and the tree T is defined by

$$\sum_{i \in A} x_i + \sum_{ij \in E_T} z_{ij} \leq |A| + \sum_{i \in V_T} (d_T(i) - 1)x_i. \tag{11}$$

Consider T as a tree rooted at t , where $t \in \bar{L}_T$ is an interior vertex. Assuming that T is not a star, it has at least one branch containing two or more vertices. For any vertex $j \in T$, denote by B_j the branch containing the minimum number of vertices among those containing at least two vertices. We let $b(T)$ be the maximum size of a branch of T when the tree is rooted in an arbitrary interior vertex. This value equals $\max_{j \in \bar{L}_T} \{|B_j|\}$. Note that Johnson et al. [17] and Mehrotra [21] introduced a family of facet-inducing inequalities, referred to as tree inequalities, that correspond to the case $A = \emptyset$ in (11). The inequalities (11) provide a facet-inducing superclass of this family.

We are ready to prove that this family of inequalities is valid and conditions that guarantee that it is facet-defining.

Theorem 8. The tree inequality (11) is valid for $P(G, k)$. Furthermore, if T is not a star, $2k - |V| - \ell + 2 \leq |A| \leq k - 2$, and $k + b(T) \leq |V| - 1$, then (11) induces a facet of $P(G, k)$.

Proof. Let $(x, z) \in P(G, k) \cap \mathbb{Z}^{|V|+|E|}$ be a feasible solution. Let $M = \{ij \in E_T : z_{ij} = 1\}$ be the subset of the tree whose edges have $z_{ij} = 1$, and let $R = \{r \in V_T : \sum_{j \in N_T(r)} z_{rj} < d_T(r)\}$. Assume first that $x_a = 1$ for every $a \in A$. Since $|V_T| = k - |A| + 1$, then R is nonempty. It suffices to show that $\sum_{ij \in M} z_{ij} \leq \sum_{i \in V(M)} (d_T(i) - 1)x_i$. As M is a forest, we will prove that $\sum_{ij \in M_t} z_{ij} \leq \sum_{i \in V(M_t)} (d_T(i) - 1)x_i$ holds for every maximal tree $M_t \subseteq M$. Consider the tree M_t as rooted in a vertex $q \in M_t$ such that $q \in R$. We know that such vertex q exists in every maximal M_t , as T is a tree, R is nonempty and M_t is maximal. Then, q has at most $d_T(q) - 1$ neighbors in $T \setminus R$ so the inequality $\sum_{i: iq \in E(T)} z_{iq} \leq (d_T(q) - 1)x_q$ holds. Now mark the edges $iq \in E(T)$ and traverse the remaining vertices in a BFS order rooted at q . Each vertex v has left at most $d_T(v) - 1$ unmarked incident edges ju such that $z_{ju} = 1$. This procedure can be repeated until no unmarked edges are left in M . As in each step the partial inequalities hold, and no edges are left, then the inequality holds. Assume now that $x_a = 0$ for some $a \in A$. Hence, $\sum_{a \in A} x_a < |A|$. If R is nonempty, we have seen that $\sum_{ij \in E_T} z_{ij} \leq \sum_{i \in V_T} (d_T(i) - 1)x_i$, so the inequality will hold. On the contrary, if R is empty, then $x_i = 1$ for every $i \in V_T$. Then, $\sum_{i \in V_T} (d_T(i) - 1)x_i = k - |A| - 1$, so the RHS of the inequality is $k - 1$. As there are at most $k - |A|$ edges $ij \in E_T$ such that $z_{ij} = 1$, the LHS of the inequality is at most $k - 1$, hence the inequality holds. Since (x, z) is an arbitrary integer point in $P(G, k)$, then (11) is valid for this polytope.

Now we switch to proving that the inequality is facet-defining. Suppose that the hypotheses of the theorem hold. Let F be the face of $P(G, k)$ defined by (11), which satisfies $\lambda^T(x, z) = \lambda_0$ for every $(x, z) \in F$. We shall prove that (λ, λ_0) is a linear combination of the coefficient vector of the tree inequality (11) and the coefficient vector of the model constraint (1b), thus showing that F is a facet of $P(G, k)$.

Claim 1. $\lambda_{z_{jl}} = 0 \forall j, l \in E \setminus E_T$. To prove this claim, we partition $E \setminus E_T$ in three subsets.

- **Claim 1a.** $\lambda_{z_{jl}} = 0 \forall j, l \in V \setminus \bar{L}_T : j, l \in E$. Consider the feasible solution $w^1 = (x, 0) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$, and the number of vertices $v \in V \setminus \bar{L}_T$ such that $w_{x_v}^1 = 1$ is exactly k . This construction is feasible since $|V| \geq 2k + 2 - |A| - \ell$. It also satisfies (11) with equality because $w_{x_a}^1 = 1$ for every vertex $a \in A$ and $w_{x_r}^1 = 0$ for every vertex r having a non-null coefficient in the RHS of the inequality. Assume, furthermore, $w_{x_j}^1 = w_{x_l}^1 = 1$, since $|A| \leq k - 2$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{jl}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{z_{jl}} = 0$.
- **Claim 1b.** $\lambda_{z_{jl}} = 0 \forall j \in A, l \in \bar{L}_T : j, l \in E$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A$, $w_{x_t}^1 = 1$ for all $t \in V_T \setminus \{s\}$, where s is an arbitrary leaf of T (i.e., $s \in L_T$), and $w_{z_{tu}}^1 = 1$ for all $tu \in E_T, t, u \neq s$. Notice that $w^1 \in F$, since the value of the LHS is $k - 1$ ($|A|$ vertices from A plus the $k - |A| - 1$ edges in T) and the value of the RHS is $|A| + \sum_{i \in V_T} (d_T(i) - 1) = |A| + 2(k - |A|) - (k - |A| + 1) = k - 1$. Assume $w_{x_j}^1 = w_{x_l}^1 = 1$ and let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{jl}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{z_{jl}} = 0$.
- **Claim 1c.** $\lambda_{z_{jl}} = 0 \forall j \notin A \cup V_T, l \in \bar{L}_T$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_a}^1 = 1$ for all vertices $a \in A$, $w_{x_j}^1 = 1, w_{x_t}^1 = 1$ for all $t \in V_T \setminus B_l, w_{z_{tu}}^1 = 1$ for all $tu \in E_T, t, u \in V_T \setminus B_l$, and the remaining vertices in the solution (if any) are chosen from $V \setminus (A \cup V_T)$. Recall that this is possible since $|V| \geq k + 1 + \max_{t \in \bar{L}_T} \{|B_t|\}$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{jl}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{z_{jl}} = 0$. Notice that if T is a star, the construction of w^1 is not feasible. As $|B_l| = 1$, by setting $w_{x_i}^1 = 1$ for every $i \in (V_T \cup A) \setminus B_l$ we already have k vertices, and thus we cannot set $w_{x_j}^1 = 1$ for a vertex $j \notin A \cup V_T$. Even more, if $i \in \bar{L}_T$ is the center of the star, then for every $w = (x, z) \in F, w_{z_{ij}}$ must be equal to zero, for every $j \notin V_T$, and thus F may not be a facet of $P(G, k)$.

Claim 2. $\lambda_{x_i} = \lambda_{x_j} \forall i, j \in V \setminus (\bar{L}_T \cup A)$. Consider the feasible solution $w^1 = (x, 0) \in F$, where $w_{x_a}^1 = 1$ for all $a \in A, w_{x_i}^1 = 1, w_{x_j}^1 = 0$, and the number of vertices $v \in V \setminus \bar{L}_T$ such that $x_v = 1$ is exactly k . Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_i}^2 = 0, w_{x_j}^2 = 1$ and keeping the remaining variables unchanged. This construction is feasible since $|V| \geq 2k + 2 - |A| - \ell$. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{x_i} = \lambda_{x_j}$.

Claim 3. (a) $\lambda_{z_{ij}} = \lambda_{z_{lm}} \forall ij, lm \in E_T$, and (b) $\lambda_{x_i} = \lambda_{x_j} - (d_T(i) - 1)\lambda_{z_{im}} \forall i \in \bar{L}_T, j \in V \setminus (A \cup \bar{L}_T), lm \in E_T$. If $|E_T| = 1$, statement (a) is trivial, so we may assume that one of i, j belongs to \bar{L}_T and, w.l.o.g., we assume $i \in \bar{L}_T$. Consider T as rooted in i . Let T_1, \dots, T_q be the subtrees obtained from $T \setminus \{i\}$ as in Fig. 1(a). As $i \in \bar{L}_T$, we know $q \geq 2$. Since $|V| \geq k + 1 + b(T)$, for each $1 \leq r \leq q, |V| \geq k + 1 + |T_r|$. This allows us to build a feasible solution $w^1 = (x, z) \in F$, in which $w_{x_a}^1 = 1$ for all $a \in A, w_{x_i}^1 = 0, w_{x_d}^1 = 1$ for all $d \in \bigcup_{r=1, r \neq u}^q T_r, w_{z_{de}}^1 = 1$ for all $de \in \bigcup_{r=1, r \neq u}^q E(T_r)$, where T_u is an arbitrary subtree, and the remaining $|T_u|$ vertices are selected from $V \setminus (A \cup V_T)$ (see Fig. 1(b)). Let j_r be the root of T_r , for $r = 1, \dots, q$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_v}^2 = 0$ for some vertex $v \in V \setminus (A \cup V_T), w_{x_t}^2 = 1$ for some leaf $t \in L_T \cap T_u$. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{x_v} = \lambda_{x_t}$. Since i, v, u , and t are arbitrary,

$$\lambda_{x_v} = \lambda_{x_t} \text{ for every } v \in V \setminus (A \cup V_T), t \in L_T. \tag{12}$$

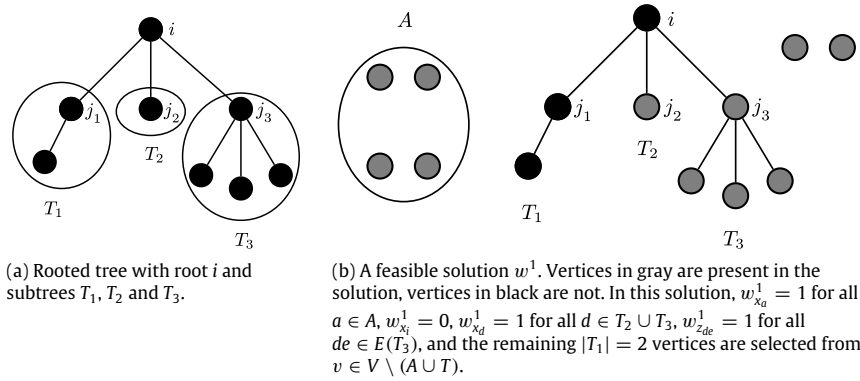


Fig. 1. Examples of the concepts involved in the proof of Claim 3.

Now, let $w^3 \in F$ be obtained from w^1 by setting $w^3_{x_v} = 0$ for some vertex $v \in V \setminus (A \cup V_T)$, $w^3_{x_i} = 1$, $w^3_{z_{jr}} = 1$ for $r = 1, \dots, q$, $r \neq u$, and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^3$, implying

$$\lambda_{x_v} = \lambda_{x_i} + \sum_{1 \leq r \leq q, r \neq u} \lambda_{z_{jr}}. \tag{13}$$

Using the same construction as before but considering a different T_l ($l \neq u$), we obtain $\lambda_{x_v} = \lambda_{x_i} + \sum_{1 \leq r \leq q, r \neq l} \lambda_{z_{jr}}$, thus concluding $\lambda_{z_{ijl}} = \lambda_{z_{iju}}$ for every $j_u, j_l \in N_T(i)$. Pivoting on each internal vertex in the path from ij to lm in T , we obtain $\lambda_{z_{ij}} = \lambda_{z_{lm}}$ and thus prove Part (a).

Notice that (13) and (12), together with Part (a) of the claim, imply $\lambda_{x_i} = \lambda_{x_j} - (d_T(i) - 1)\lambda_{z_{im}}$, for $i \in \bar{L}_T, j \in V \setminus (A \cup \bar{L}_T)$, $lm \in E_T$.

Claim 4. $\lambda_{x_a} = \lambda_{x_j} + \lambda_{z_{il}} \forall a \in A, j \in V \setminus (A \cup \bar{L}_T), il \in E_T$. By Claims 3 and 2, we can choose a convenient edge il and vertex j , since the value $\lambda_{z_{il}}$ is the same for all the edges il of T and the value λ_{x_j} is the same for all the vertices j of $V \setminus (A \cup \bar{L}_T)$. So, let $l \in L_T$, and consider the feasible solution $w^1 = (x, z) \in F$, where $w^1_{x_a} = 1$ for all vertices $a \in A$, $w^1_{x_t} = 1$ for all $t \in V_T \setminus \{l\}$. Let $w^2 \in F$ be obtained from w^1 by setting $w^2_{x_a} = 0$ for some $a \in A$, $w^2_{x_i} = 1$, $w^2_{z_{il}} = 1$ where $il \in E_T$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{x_a} = \lambda_{x_i} + \lambda_{z_{il}}$.

With these claims, we now define $\alpha := \lambda_{z_{ij}}$ for any $ij \in E_T$. Note that, by Claim 3a, the choice of ij does not alter the value of α . We define $\beta := \lambda_{x_i}$ for $i \in V \setminus (A \cup \bar{L}_T)$. Again, this is well defined by Claim 2. Claim 4 implies that $\lambda_{x_a} = \alpha + \beta$ for each $a \in A$, Claim 3b implies that $\lambda_{x_i} = \beta - (|d_T(i)| - 1)\alpha$ for each $i \in \bar{L}_T$, and Claim 1 implies that $\lambda_{z_{ij}} = 0$ for every $ij \in E \setminus E_T$. We conclude that λ is a linear combination of the coefficient vector π of the tree inequality (11) and the coefficient vector π' of the model constraint (1b); i.e., $\lambda = \alpha\pi + \beta\pi'$. Let π_0 and π'_0 be the independent terms of (11) and (1b), respectively. The calculation in (3) can be replicated for this case, concluding that (λ, λ_0) is a linear combination of (π, π_0) and (π', π'_0) and, therefore, (11) induces a facet of $P(G, k)$.

Finally, if T is a star of center i , then the tree inequality (11) is dominated by (2), the generalized neighborhood inequality associated with the same set A and the vertex i , hence (11) does not induce a facet of $P(G, k)$ in this case. \square

The separation problem for the tree inequalities can be defined as follows.

TREE INEQUALITIES SEPARATION

Instance: A point $w = (x, z) \in P_{LP}(G, k)$ and an integer k .

Question: Does w violate some tree inequality?

The problem TREE INEQUALITIES SEPARATION is NP-complete because it involves finding a minimal dependent set that minimizes a linear function, which is an NP-hard problem [17]. We now outline an alternative, self-contained proof of this result.

Theorem 9. TREE INEQUALITIES SEPARATION is NP-complete.

Proof Sketch. We go along the same lines as in the proof of Theorem 7, by showing that r -CARDTREE can be reduced to TREE INEQUALITIES SEPARATION. Given an instance (H, r, p) , where $H = (V_H, E_H)$ is an edge-weighted graph with non-negative edge weights $w_H : E_H \rightarrow [0, p]$, we construct a solution $w = (x, z)$ in the following way:

- Let $k = r$.
- Set $w_{x_i} = k/n$ for every $i \in V_H$.
- Set $w_{z_{ij}} = (k/n)(1 - w_H(ij)/p)$ for every $ij \in E_H$.

It can be easily verified that $0 \leq w_{z_{ij}} \leq k/n = w_{x_i}$ for every $ij \in E$. In this setting, it can be proved that $\sum_{ij \in T} w_H(ij) < p \Leftrightarrow \sum_{ij \in E_T} w_{z_{ij}} > \sum_{i \in V_T} (d_T(i) - 1)x_i$. In addition, any affirmative instance of TREE INEQUALITIES SEPARATION must have $r = k - 1$ edges, so it must be a tree, and hence $|A|$ must be empty. \square

Since the separation problem for the tree inequalities is NP-complete, we propose a heuristic procedure for separating these inequalities (see Algorithm 3). The general idea is to build maximum-size trees that violate the inequality. If the tree has $k + 1$ vertices, due to the construction of the directed graph G_H , it will violate the inequality. Otherwise, we check if we can add a set A such that the inequality associated with both A and T is violated.

Algorithm 3 Heuristic for the separation of the tree inequalities

Input: A point $(x, z) \in P_{LP}(G, k)$.

Output: A set of violated tree inequalities (which may be empty)

Construct the directed graph $G_H = (V_H, E_H)$ in the following way:

$$V_H = \{i \in V : 0 < x_i\},$$

$$E_H = \{i \rightarrow j : ij \in E, 0 < z_{ij}, z_{ij} = x_i\}.$$

Set $S = \emptyset$.

for $v \in V_H$ **do**

Using BFS build a tree T starting from v until it has $k + 1$ vertices or no more vertices can be added to T .

if T has $k + 1$ vertices **then**

The inequality associated with the tree T and $A = \emptyset$ is violated by x_v .

Add the inequality to S .

else

Set $r = |V(T)|$.

Let A be the set of $(k - r + 1)$ vertices in $V_H \setminus T$ such that $\sum_{a \in A} x_a$ is maximum.

if $(k - r + 1) - \sum_{a \in A} x_a < x_v$ **then**

The inequality associated with the tree T and the vertex set A is violated by $x_v + \sum_{a \in A} x_a - (k - r + 1)$.

Add the inequality to S .

end if

end if

end for

Return S

3.5. Disjoint matching inequalities

We now consider a more advanced structure defined by a matching and a tree.

Definition 5. Let $M \subseteq E$ be a matching such that $0 < |M| < (k - 1)/2$ and let $T = (V_T, E_T) \subseteq G$ be a tree on $k - 2|M| + 1$ vertices from $V \setminus V(M)$. A disjoint matching inequality associated with the tree T and the matching M is defined by

$$\sum_{ij \in M} z_{ij} + \sum_{ij \in E_T} z_{ij} \leq |M| + \sum_{i \in V_T} (d_T(i) - 1)x_i. \tag{14}$$

This family of inequalities is valid and facet-defining under the conditions below. The proof is similar to that of [Theorem 8](#) and is deferred to the [Appendix](#).

Theorem 10. The disjoint matching inequality (14) is valid for $P(G, k)$. Furthermore, if $2k < |V| + 2|M| + \ell$, and $k + b(T) \leq |V| - 1$, then (14) induces a facet of $P(G, k)$.

It can be seen that the separation problem related to this family of inequalities is NP-complete. We omit the proof because it follows the same lines as that of [Theorem 9](#). Moreover, one can provide a heuristic that addresses the separation problem using the ideas of [Algorithm 3](#).

4. The branch and cut algorithm

This section briefly describes the implementation of a branch and cut (B&C) algorithm based on the polyhedral results presented in [Section 3](#). We employ the formulation *MIP1* described in (1a)–(1f), strengthened with all neighborhood inequalities associated with vertices of degree at least k . Recall that the neighborhood inequality associated with a vertex i is a particular case of the generalized neighborhood inequality associated with the vertex i and the set $A = \emptyset$. [Theorem 2](#) proves that it is facet-inducing if $|N(i)| \geq k$, and hence the restriction on the degree used for the strengthening.

The B&C procedure dynamically adds cuts generated from generalized neighborhood inequalities (2), matching inequalities (7), forest inequalities (9), tree inequalities (11), and disjoint matching inequalities (14). To separate these inequalities, we implemented the algorithms presented in the corresponding sections. For the matching inequalities, we implemented both the exact algorithm described in the proof of [Theorem 5](#) and a heuristic that constructs the maximum weight matching using a greedy algorithm.

To exploit the potential structure in the fractional solutions at each node in the B&C tree, we also implemented a standard *primal heuristic* based on solution rounding. We construct a feasible solution from each fractional one by selecting the vertices associated with the k highest x -variables in the fractional solution, and by setting the z -variables accordingly.

Since the x -variables are sufficient for defining a feasible solution, we branch on these variables only. When performing a branching step, we select a fractional x -variable and create two children vertices that force this variable to be 0 and 1, respectively.

4.1. Computational experiments

The B&C procedure explained above was implemented under the ABACUS framework [16], using CPLEX 9.1 to solve the linear relaxations. The experiments were performed on a Core™2 Duo CPU, running at 1.5 GHz with 1 Gb of RAM memory and a 1 h time limit.

We divided the experiments in two phases. During the first phase, we focused on tuning the implementation of our B&C procedure by selecting the families of inequalities that achieved the best performance. During the second phase, we tested our B&C algorithm against CPLEX. We used CPLEX with default parameters in all cases to make the comparison fair.

For the computational study we selected instances of three different classes: random instances, well-known benchmark instances and real-world graphs. To generate random instances of different sizes and edge densities, we employed the NetworkX package [14] to produce two types of random graphs:

- *Erdős–Rényi random graphs* – $ER(n, p)$: These is the simplest kind of random graph, where one fixes the number of vertices and its density. Starting with an empty graph of n vertices, we connect each dyad with probability p , using independent draws from a Bernoulli distribution with parameter p [11].
- *Barabási–Albert random graphs* – $BA(n, m)$: Besides using ER random graphs, since the MESP frequently appears in social networks analysis, we also selected graphs with topological structures that are similar to those from the real world. Specifically, we generate *scale-free* graphs [2]. To construct these graphs, we start with a graph consisting of an isolated vertex, and successively add vertices one at a time until a total of n vertices are added. At each step, we connect a new vertex to m of the existing vertices using *preferential attachment*, i.e., the probability of selecting a given existing vertex is proportional to its current degree [5]. The result is a *scale-free* network that has a power-law degree distribution.

For the second group, we selected some instances from the 1993 DIMACS challenge [18]. Although these instances were submitted to the challenge with the goal of finding maximum cliques, the structure of both problems is similar enough to make the library very appropriate for our purposes. Finally, we also included some instances representing Bilateral Investment Treaties (BITs) that were introduced in [24]. These networks are snapshots of the world taken at different times. Here, vertices represent countries and an edge is included whenever two countries signed a BIT by the date of the snapshot. We generated graphs for years 1990, 1995, 2000, 2005 and 2007 to study the evolution of BITs in the world.

A first conclusion that can be drawn from our experiments is that it pays off to add the neighborhood inequalities to the initial formulation. Strengthening the formulation in this way results in much tighter dual bounds. This was evidenced by a reduction of execution time ranging between 40% and 90%. In addition, the strengthening allowed us to solve larger instances because it prevented the computer from running out of memory in many cases because of the resulting pruning of the branch and bound tree.

We also explored the effectiveness of different combinations of families of valid inequalities. To benchmark our cut generation procedure, we implemented a simple branch and bound (B&B) solution under ABACUS. To this end, we created instances of Erdős–Rényi and Barabási–Albert random graphs with different numbers of vertices and edge densities. We created 25 instances for each set of parameters, tested the B&C algorithm with each family of inequalities alone, and compared the execution times with those of the B&B algorithm. [Table 1](#) illustrates some of these results.

Although all the families alone and most of the combinations presented a considerable improvement with respect to the plain branch and bound algorithm, the best results were achieved when separating the generalized neighborhood inequalities at each node of the B&B tree. Only one execution of Algorithm 1 is performed at each node, and the obtained inequalities are added as cuts to the current subproblem (although they are valid globally). Making use of additional families after having added cuts from this family proved ineffective because the extra cuts do not offset the time needed to identify the violated inequalities.

According to our experiments, the execution times were highly influenced by the following three factors:

- *The structure of the instance*. The execution times were much higher for ER graphs than for BA graphs, keeping the number of vertices, the edge density and the value of k fixed.
- *The edge density*. As the edge density of the instance increases, so does the execution time. Bigger instances require more storage memory and handling them requires more overhead. Nevertheless, this could be a limitation of our implementation.

Table 1

Comparison between ABACUS B&B and ABACUS B&C enhanced with the GNI, tree, and forest inequalities. Times are in seconds and correspond to averages over 25 random instances generated with the same parameters. The results for the matching inequalities are omitted since their performance was poor. Times are reported in seconds, and ** specifies that the solver ran out of memory.

Instance	m	k	B&B (time)	B&C (time)		
				GNI	Tree	Forest
ER(30, 0.2)	93	6	20.89	4.72	22.15	30.34
ER(30, 0.4)	173	6	332.99	12.41	173.53	98.00
ER(30, 0.6)	254	6	**	3.19	14.21	12.85
ER(30, 0.8)	354	6	**	2.74	7.69	9.26
ER(40, 0.2)	129	8	402.62	10.43	47.92	38.97
ER(40, 0.4)	318	8	**	58.85	218.89	128.12
ER(40, 0.6)	462	8	**	4.40	15.58	15.45
ER(40, 0.8)	606	8	**	3.64	5.92	7.30
ER(40, 0.2)	129	8	402.62	10.43	47.92	38.97
ER(40, 0.2)	129	10	407.99	26.05	64.68	63.00
ER(40, 0.2)	129	12	515.58	59.59	170.69	132.23
ER(40, 0.2)	129	14	401.48	91.66	76.46	186.24
BA(30, 2)	56	6	8.02	10.43	36.85	32.51
BA(30, 4)	104	6	12.33	18.85	69.33	64.29
BA(30, 6)	144	6	16.89	4.40	16.21	18.31
BA(30, 8)	175	6	110.16	3.64	9.57	10.28
BA(40, 2)	76	8	1.57	0.77	0.58	0.63
BA(40, 4)	144	8	150.76	3.52	10.11	7.73
BA(40, 12)	336	8	**	8.10	26.50	20.68
BA(40, 18)	459	8	**	3.95	17.28	15.73
BA(40, 4)	144	8	150.76	3.52	10.11	7.73
BA(40, 4)	144	10	183.91	5.83	21.26	16.38
BA(40, 4)	144	12	231.82	12.37	37.88	35.24
BA(40, 4)	144	14	217.76	9.28	29.94	31.65

- *The relation between k and the clique number $\omega(G)$.* (The clique number of a graph is the cardinality of its largest clique.) The case of $k \leq \omega(G)$ can be solved efficiently, even for difficult instances, because in that case the problem consists of identifying the vertices that compose the clique of cardinality k . However, as k increases and goes over the clique number, the execution time increases.

We tested the following policies for selecting the branching variable: the first vertex that is not fixed, a random vertex, the vertex with most neighbors, and the vertex with the highest value in the LP relaxation of the current B&B node. The best performance was obtained with the last of those policies. The node selection rule was set to be DFS. Finally, the total time spent in adding the inequalities was between 3% and 20% of the overall CPU time, depending on the size of the instance. When the instance has a high number of edges, fewer nodes of the B&B tree are explored, hence the relative time devoted to the separation routines is smaller.

To put our results in perspective, Mehrotra found that star inequalities alone reduced approximately 47% of the LP gap (the average of Tables 1–3 in [21]). In our case, GNI inequalities, a superclass of Mehrotra's star inequalities, provide a reduction of approximately 73.4% of the computing time with respect to pure B&B. (Note that this is an understatement because the average is taken among instances for which the pure B&B algorithm finished, which are relatively easier.) For the case of tree inequalities, Mehrotra achieves approximately a 64% reduction of the LP gap, while our tree inequalities (a superclass of Mehrotra's) achieve a 13.6% reduction of computing time. Our conclusion is that the generalization of neighborhood inequalities into the much wider class GNI, together with the complexity of its separation procedure (Algorithm 1), provide quick and effective cuts that allow us to accelerate the branching procedure significantly.

4.2. Comparison with CPLEX

We tested our implementation by comparing its performance against the running times obtained by CPLEX 9.1 with default parameters. Table 2 shows both the running times (in seconds) for CPLEX and for our implementation. The table displays two groups of instances: the group on top contains easy instances for which $k \leq \omega(G)$, while the second group contains the harder instances for which $k > \omega(G)$. In all cases we selected k to be close to the clique number, either below or above, to make the instances interesting but also not extremely hard. Note that both groups contain the same graphs, the only change is the value of k . In the cases when we could not achieve optimality within the time limit, we report the duality gap.

It can be observed in the table that CPLEX declared optimality only in a few cases. Our procedure could solve all but one instance to optimality in the easy group, but also had a hard time solving the hardest instances in the second group. Nevertheless, even for instances not solved to optimality, the gaps obtained by our procedure were orders of magnitude better than those of CPLEX.

Table 2
Comparison between CPLEX and our procedure. Times are reported in seconds, and ** specifies that the running time exceeded the time limit of 1 h.

Instance	n	m	$\omega(G)$	k	LP relax. root node	CPLEX		B&C	
						Time	Gap (%)	Time	Gap (%)
Easy instances									
ER(100, 0.2)	100	1,021	5	5	51.05	**	47.67	10.53	–
ER(200, 0.1)	200	2,016	4	4	40.32	**	275.45	75.17	–
ER(200, 0.2)	200	4,022	5	5	100.55	**	716.30	503.75	–
BA(100, 10)	100	900	8	8	72.00	593.66	–	11.71	–
BA(200, 10)	200	1,900	8	8	76.00	**	106.52	39.37	–
BA(200, 15)	200	2,392	8	8	95.68	**	113.70	142.74	–
c-fat200-1	200	1,534	12	12	92.04	432.00	–	46.63	–
c-fat200-2	200	3,235	24	12	194.10	**	138.41	88.00	–
hamming6_4	64	704	4	4	44.00	704.88	–	4.61	–
hamming6_2	64	1,824	32	30	855.00	**	73.81	757.36	–
johnson8-2-4	28	210	4	4	30.00	2.20	–	1.03	–
johnson8-4-4	70	1,855	14	14	371.00	**	204.47	303.92	–
keller4	171	9,435	11	8	441.40	**	474.41	1587.79	–
san200_07_1	200	13,930	30	30	2098.50	**	415.78	**	4.68
MANN_a9	45	918	18	15	306.00	**	84.66	22.26	–
brock200_1	200	14,834	21	10	741.70	**	–	407.37	–
BITs 1990	113	447	5	5	19.77	615.37	–	3.24	–
BITs 1995	158	1,192	7	7	52.81	**	44.68	12.39	–
BITs 2000	173	1,994	12	12	138.31	**	88.96	84.61	–
BITs 2005	179	2,460	14	13	178.65	**	92.26	97.32	–
BITs 2007	179	2,529	15	14	197.79	**	94.72	105.64	–
Hard instances									
ER(100, 0.2)	100	1,021	5	7	71.47	**	89.72	1108.02	–
ER(200, 0.1)	200	2,016	4	7	70.56	**	211.14	**	6.66
ER(200, 0.2)	200	4,022	5	7	140.77	**	603.65	3061.24	–
BA(100, 10)	100	900	8	10	90.00	640.46	–	236.43	–
BA(200, 10)	200	1,900	8	10	95.00	**	64.10	2494.92	–
BA(200, 15)	200	2,392	8	10	119.60	**	119.39	2831.53	–
c-fat200-1	200	1,534	12	14	107.38	658.60	–	**	3.74
c-fat200-2	200	3,235	24	24	388.20	**	38.63	1155.35	–
hamming6_4	64	704	4	7	77.00	**	158.21	**	6.35
hamming6_2	64	1,824	32	35	997.50	**	54.30	**	4.90
johnson8-2-4	28	210	4	8	60.00	60.73	–	625.32	–
johnson8-4-4	70	1,855	14	16	424.00	**	192.81	**	7.14
keller4	171	9,435	11	13	717.28	**	354.59	**	4.60
san200_07_1	200	13,930	30	32	2228.80	**	410.80	**	4.80
MANN_a9	45	918	18	18	367.20	**	64.29	**	1.32
brock200_1	200	14,834	21	25	1854.25	**	611.50	**	15.40
BITs 1990	113	447	5	7	27.69	2274.45	–	918.23	–
BITs 1995	158	1,192	7	8	60.35	**	61.12	2936.26	–
BITs 2000	173	1,994	12	14	161.36	**	73.41	**	0.12
BITs 2005	179	2,460	14	15	206.14	**	79.94	**	0.09
BITs 2007	179	2,529	15	16	226.05	**	85.10	**	0.08

To further compare the two solution procedures, Table 3 presents the optimal (or best when the time limit was reached) solution and the number of explored nodes in the B&B tree for the group of hard instances. The cuts allowed us to significantly prune the B&B tree, and consequently our method always returned solutions that dominated those of the standard procedure. While CPLEX explored thousands of nodes within the 1 h time limit, our algorithm explored less than 5% of this amount. This provides evidence of the power of the generated cuts. On the negative side, a limiting factor of our implementation is the overhead imposed by the communication between ABACUS and CPLEX, whereas CPLEX does the B&C procedure internally. This overhead, plus the extra time needed to generate cuts, limited the number of nodes that could be explored within the time limit. For this reason, although our method can find good solutions relatively quickly, sometimes it cannot prove optimality (e.g., with instance cfat200-1). The overhead also explains why in some cases when instances are small (e.g., instances cfat200-1 and johnson8-2-4), CPLEX outperforms our method.

Finally, Table 4 shows how the performance of both our CPLEX and our B&C procedures varies as we change the value of k , for the DIMACS benchmark instances we considered above. These experiments were made for values of k greater than ω . As expected, the gap in our method increases as k increases, but the B&C still outperforms CPLEX in most cases.

Going back to the original motivation of computing quasi-cliques to determine the cohesiveness of groups of countries with respect to their signature of Bilateral Investment Treaties, looking at the output in Table 3, we can see how in the

Table 3

Primal bounds and number of nodes in the enumeration trees with CPLEX and B&C for the group of hard instances with $k > \omega(G)$. The column ‘Opt?’ displays a check mark if optimality was proved.

Instance	n	m	k	CPLEX			B&C		
				Best sol.	#Sub	Opt?	Best sol.	#Sub	Opt?
ER(100, 0.2)	100	1,021	7	17	44,300		18	391	✓
ER(200, 0.1)	200	2,016	7	15	149,200		15	221	
ER(200, 0.2)	200	4,022	7	15	44,800		19	161	✓
BA(100, 10)	100	900	10	43	65,200	✓	43	147	✓
BA(200, 10)	200	1,900	10	36	107,600		41	239	✓
BA(200, 15)	200	2,392	10	34	98,200		43	267	✓
c-fat200-1	200	1,534	14	79	27,900	✓	79	461	
c-fat200-2	200	3,235	24	244	37,800		263	19	✓
hamming6_4	64	704	7	15	502,700		16	2551	
hamming6_2	64	1,824	35	554	303,000		577	605	
johnson8-2-4	28	210	8	21	144,000	✓	21	2417	✓
johnson8-4-4	70	1,855	16	104	200,800		112	631	
keller4	171	9,435	13	63	92,100		63	35	
san200_07_1	200	13,930	32	388	4,100		415	17	
MANN_a9	45	918	18	149	1023,200		151	1615	
brock200_1	200	14,834	25	246	3,600		260	13	
BITs 1990	113	447	7	18	897,100	✓	18	331	✓
BITs 1995	158	1,192	8	25	203,200		27	286	✓
BITs 2000	173	1,994	14	86	135,200		90	498	
BITs 2005	179	2,460	15	99	131,800		104	451	
BITs 2007	179	2,529	16	111	127,600		119	439	

Table 4

Comparison between CPLEX and our procedure for different values of the parameter $k \geq \omega(G)$ for the considered DIMACS benchmark instances. Times are reported in seconds, and ** specifies that the running time exceeded the time limit of 1 h.

Instance	n	m	$\omega(G)$	k	LP relax. root node	CPLEX		B&C	
						Time	Gap (%)	Time	Gap (%)
c-fat200-1	200	1,534	12	13	99.71	598.73	–	2825.23	–
				14	107.38	658.60	–	**	3.74
				15	115.05	764.32	–	**	4.32
c-fat200-2	200	3,235	24	24	388.20	**	38.63	1155.35	–
				25	404.37	**	39.04	3211.87	–
				26	420.55	**	39.46	**	2.73
hamming6_4	64	704	4	5	55.00	**	55.42	1495.62	–
				6	66.00	**	121.36	3182.34	–
				7	77.00	**	158.21	**	6.35
hamming6_2	64	1,824	32	33	940.50	**	68.23	2937.45	–
				34	969.00	**	59.24	**	1.63
				35	997.50	**	54.30	**	4.90
johnson8-2-4	28	210	4	6	45.00	10.24	–	87.45	–
				7	52.50	31.36	–	249.44	–
				8	60.00	60.73	–	625.32	–
johnson8-4-4	70	1,855	14	15	397.50	**	198.45	**	1.99
				16	424.00	**	192.81	**	7.14
				17	450.50	**	188.23	**	9.42
keller4	171	9,435	11	12	662.10	**	427.55	**	2.34
				13	717.28	**	354.59	**	4.60
				14	772.45	**	328.62	**	5.82
san200_07_1	200	13,930	30	31	2159.15	**	412.27	**	4.95
				32	2228.80	**	410.80	**	4.80
				33	2298.45	**	408.05	**	5.67
MANN_a9	45	918	18	18	367.20	**	64.29	**	1.32
				19	387.60	**	59.34	**	2.04
				20	408.00	**	56.81	**	3.97
brock200_1	200	14,834	21	23	1705.91	**	704.56	**	2.64
				24	1780.08	**	672.30	**	5.27
				25	1854.25	**	611.50	**	15.40

last 20 years the world has become more cohesive because quasi-cliques are larger. In 1990 there was a quasi-clique of cardinality 7 and density 86%. This means that there is a group of 7 countries out of which 86% of all combinations of countries signed treaties with each other. For 1995, 2000, 2005 and 2007, the maximum densities for quasi-cliques with cardinalities 8, 14, 15 and 16 are 96%, 99%, 99% and 99%, respectively. In other words, except for the instance of 1990 that is missing 3

treaties, our best solution consists of groups of countries that signed treaties with each other except for just one pair. The speed of our procedure facilitates the construction of pareto-curves for the cardinality and density of quasi-cliques [24].

5. Concluding remarks

We have explored the maximum edge subgraph problem with a polyhedral combinatorics perspective, presenting a polyhedral study of the integer programming formulation of [6]. We have introduced five families of valid inequalities for the associated polytope, which show that the polytope admits facets with a simple combinatorial structure. For each family of inequalities, we have determined the computational complexity of the associated separation problem and, based on this complexity, we have presented either an exact or a heuristic separation procedure. Based on these families of cuts, we have implemented a branch and cut algorithm for this problem. The objective of our computational study was to test the strength of our inequalities. Although we could not solve all the instances to optimality, we have obtained promising results, in most cases improving the results given by CPLEX, and obtaining much smaller gaps.

As mentioned in the Introduction, the detection of quasi-cliques was an integral part of the analysis of the network of bilateral investment treaties that appeared in [24]. In that study, we detected highly cohesive groups of countries by looking at quasi-cliques, which was deemed more appropriate than having used regular cliques because for unexplained reasons it is possible that some countries in the same bloc do not sign treaties with each other. The machinery presented in the current article allowed us to perform the necessary computations. For example, the clique number of the network of treaties signed by 2005 is 14, while the solution of the MESP with $k = 18$ and the same network is a group of countries that misses only 5 out of 153 edges (3.3%) to become a clique. Raising k to 22 shows that the most cohesive group of countries misses 20 out of 231 edges (8.7%) to become a clique.

We conjecture that some of the families introduced in this work can be generalized. In particular, we believe that the tree inequalities and the disjoint matching inequalities can be generalized into a single family. It would also be interesting to implement the branch and cut algorithm directly within CPLEX to perform a more direct comparison to a pure branch and bound algorithm.

Acknowledgments

We wish to thank the anonymous referees for their valuable comments on the paper. This work was partially supported by ANPCyT PICT-2007-00518 and 00533, by UBACyT Grants X069, X606, 20020100100980 and 20020090300094, by CONICET PIP 112-200901-00178 (Argentina), and by the Columbia University Center for International Business Education and Research (USA). An extended abstract of this paper was presented at LAGOS'09 [7].

Appendix A. Additional proofs

A.1. Proof of Theorem 4

Here, we prove a result that implies Theorem 4. Indeed, notice that the last condition of this result does not impose any restriction when $k \leq |V| - 3$ because in that case the condition is automatically false.

Theorem 11. *If $k - 2|B| + 1 \leq |A| \leq k - 1$ then the matching inequality (7) is facet-defining, unless $|V| - 1 < |A| + 2|B| < k + 3$.*

Proof. Let F be the face of $P(G, k)$ defined by (7) and suppose $\lambda^T(x, z) = \lambda_0$ for every $(x, z) \in F$. We shall prove that (λ, λ_0) is a linear combination of the coefficient vector of the matching inequality (7) and the coefficient vector of the model constraint (1b), thus showing that F is a facet of $P(G, k)$. Notice that every solution $(x, z) \in F$ has $\sum_{i \in V \setminus (A \cup V(B))} x_i \leq 1$ because when (7) is tight we must have $|A| - 1 \leq r \leq |A|$. Furthermore, if $i, j \in V \setminus (A \cup V(B))$ then $ij \notin E$, since B is a maximal matching.

Claim 1. $\lambda_{z_{jq}} = 0$ for every $jq \in E \setminus B$. Set M to be a subset of cardinality $(k - |A| - 1)/2$ of the matching B , noting that the first condition in our hypothesis implies that $0 \leq |M| < |B|$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_a}^1 = 1$ for every $a \in A$, $w_{z_{bc}}^1 = 1$ for every $bc \in M$, $w_{x_b}^1 = 1$ for every $b \in V(M)$, and $w_{x_i}^1 = 1$ for exactly one vertex $i \in V \setminus (A \cup V(M))$. This construction is well defined because there must be some vertex outside $A \cup V(M)$. Using this construction we can maneuver to have $w_{x_j}^1 = w_{x_q}^1 = 1$. We set w^2 equal to w^1 except for the coordinate $w_{z_{jq}}^2 = 1$, which was zero in w^1 . Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{z_{jq}} = 0$.

Claim 2. $\lambda_{x_i} = \lambda_{x_j}$ for every $i, j \in V \setminus A$. Consider the feasible solution $w^1 = (x, z) \in F$ as in Claim 1, making sure that vertex $j \notin V(M)$. This is possible because the last condition of the hypothesis implies that there are at least three vertices outside $A \cup V(M)$ (see the end of this paragraph). Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_i}^2 = 0$, $w_{x_j}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{x_i} = \lambda_{x_j}$. If there are more than two edges in $B \setminus M$, this claim is valid for any two vertices in $V \setminus A$. Now, if there is exactly one edge in $B \setminus M$, since there must be at least one vertex outside $A \cup V(B)$, this claim proves that all vertices in $V(B)$ have a value of λ equal to that of any of the unmatched vertices.

Claim 3. $\lambda_{z_{ql}} = \lambda_{z_{ij}}$ for every $ql, ij \in B$. Consider the feasible solution $w^1 = (x, z) \in F$ as in Claim 1, making sure that $ql \in M$ and $ij \notin M$. Define $w^2 \in F$ similarly but switching the roles of the two edges. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$. Looking at the coordinates where solutions differ, we obtain $\lambda_{x_q} + \lambda_{z_{ql}} = \lambda_{x_j} + \lambda_{z_{ij}}$. As q, j are vertices in $V \setminus A$, by Claim 2 we know $\lambda_{x_q} = \lambda_{x_j}$, and thus we conclude that $\lambda_{z_{ql}} = \lambda_{z_{ij}}$.

Claim 4. $\lambda_{x_a} = \lambda_{x_j} + \lambda_{z_{ij}}$ for every $a \in A$ and every $ij \in B$. Consider the feasible solution $w^1 = (x, z) \in F$ as in Claim 1, making sure that $ij \notin M$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_a}^2 = 0, w_{x_j}^2 = 1, w_{z_{ij}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying that $\lambda_{x_a} = \lambda_{x_j} + \lambda_{z_{ij}}$.

We define $\alpha = \lambda_{z_{ij}}$ for any $ij \in B$. Note that, by Claim 3, the choice of ij does not alter the definition of α . We define $\beta = \lambda_{x_i}$ for any $i \in V \setminus A$. Again, by Claim 2, the definition of β does not depend on the choice of i . From the definition of α and β we obtain $\lambda_{x_a} = \alpha + \beta$ by Claim 4 and $\lambda_{z_{jq}} = 0$ for every $jq \in E \setminus B$ by Claim 1. We conclude that λ is a linear combination of the coefficient vector of the matching inequality (7) and the coefficient vector of the model constraint (1b). A similar calculation as in the proof of Theorem 2 can be given to show that λ_0 can be written as the same combination of the independent terms of (7) and (1b), hence (7) induces a facet of $P(G, k)$. \square

A.2. Proof of Theorem 10

We prove that the family of disjoint matching inequalities is facet defining. The proof of its validity is very similar to Theorem 8 and is not included. We use the same notations for leaves, interior vertices, and the maximum size of a connected component of trees as in Section 3.4, when we worked with tree inequalities.

Proof. Suppose that the hypotheses of the theorem hold. Let F be the face of $P(G, k)$ defined by (14), which satisfies $\lambda^T(x, z) = \lambda_0$ for every $(x, z) \in F$. We shall prove that (λ, λ_0) is a linear combination of the coefficient vector of the disjoint matching inequality (14) and the coefficient vector of the model constraint (1b), thus showing that F is a facet of $P(G, k)$.

Claim 1. $\lambda_{z_{rs}} = 0 \forall rs \in E \setminus (E_T \cup M)$. To prove this claim, we partition $E \setminus (E_T \cup M)$ in three subsets.

- **Claim 1a.** $\lambda_{z_{rs}} = 0 \forall r, s \in V \setminus \bar{L}_T : rs \in E \setminus M$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_m}^1 = 1$ for all $m \in V(M), w_{z_{ij}}^1 = 1$ for all $ij \in M$, and the number of vertices $v \in V \setminus \bar{L}_T$ such that $w_{x_v}^1 = 1$ is exactly k . This construction is feasible since $|V| > 2k - 2|M| - \ell$. This construction also satisfies (14) with equality, since $w_{z_{ij}}^1 = 1$ for every edge $ij \in M$ and $w_{x_i}^1 = 0$ for every vertex i having a non-null coefficient in the RHS of the inequality. Assume, furthermore, $w_{x_r}^1 = w_{x_s}^1 = 1$, since $2|M| < k - 1$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{rs}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{z_{rs}} = 0$.
- **Claim 1b.** $\lambda_{z_{rs}} = 0 \forall r, s \in V(M) \cup V_T : rs \in E \setminus (E_T \cup M)$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_m}^1 = 1$ for every $m \in V(M), w_{z_{ij}}^1 = 1$ for every $ij \in M, w_{x_l}^1 = 0$ for one leaf $l \in L_T, w_{x_t}^1 = 1$ for all $t \in V_T \setminus \{l\}$ and $w_{z_{tu}}^1 = 1$ for all $tu \in E(T \setminus \{l\})$. Notice that $w^1 \in F$, since the value of the LHS is $k - |M| - 1$ (i.e., $|M|$ edges from M plus the $k - 2|M| - 1$ edges in $T \setminus \{l\}$) and the value of the RHS is also $k - |M| - 1$. Assume $w_{x_r}^1 = w_{x_s}^1 = 1$ and let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{rs}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{z_{rs}} = 0$ as long as $r, s \neq l$. As l in an arbitrary leaf of T , and every tree has at least two leaves, the claim follows.
- **Claim 1c.** $\lambda_{z_{rs}} = 0 \forall r \notin V(M) \cup V_T, s \in \bar{L}_T$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{z_{ij}}^1 = 1$ for exactly $|M| - 1$ edges $ij \in M, w_{x_i}^1 = 1$ for $i \in V(M)$ if and only if $w_{z_{ij}}^1 = 1$ for $ij \in M, w_{x_t}^1 = 1$ for all $t \in V_T$ and $w_{z_{tu}}^1 = 1$ for all $tu \in E_T$. Set $w_{x_r}^1 = 1$ to complete the k vertices in the solution. Let $w^2 \in F$ be obtained from w^1 by setting $w_{z_{rs}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{z_{rs}} = 0$. Notice that, in this case, it is possible for T to be a star.

Claim 2. $\lambda_{x_r} = \lambda_{x_s} \forall r, s \in V \setminus \bar{L}_T$. We prove it for two special cases below. The claim follows by transitivity because $V \setminus (V(M) \cup V_T)$ is nonempty.

- **Claim 2a.** $\lambda_{x_r} = \lambda_{x_s} \forall r, s \in V \setminus (V(M) \cup \bar{L}_T)$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{x_m}^1 = 1$ for all $m \in V(M), w_{z_{ij}}^1 = 1$ for all $ij \in M, w_{x_r}^1 = 1, w_{x_s}^1 = 0$ and the number of vertices $v \in V \setminus \bar{L}_T$ such that $w_{x_v}^1 = 1$ is exactly k . This construction is feasible since $|V| > 2k - 2|M| - \ell$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{w_{x_r}}^2 = 0, w_{w_{x_s}}^2 = 1$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{x_r} = \lambda_{x_s}$.
- **Claim 2b.** $\lambda_{x_r} = \lambda_{x_s} \forall r \in V \setminus (V(M) \cup V_T), s \in V(M)$. Consider the feasible solution $w^1 = (x, z) \in F$, where $w_{z_{ij}}^1 = 1$ for the $|M| - 1$ edges $ij \in M$ with $i, j \neq s, w_{x_i}^1 = 1$ for $i \in V(M)$ if and only if $w_{z_{ij}}^1 = 1$ for $ij \in M, w_{x_t}^1 = 1$ for all $t \in V_T, w_{z_{tu}}^1 = 1$ for all $tu \in E_T$, and $w_{x_r}^1 = 1$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{w_{x_r}}^2 = 0, w_{w_{x_s}}^2 = 1$, and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda_0 = \lambda^T w^2$, implying $\lambda_{x_r} = \lambda_{x_s}$.

Claim 3. (a) $\lambda_{z_{ij}} = \lambda_{z_{lm}} \forall ij, lm \in E_T$, and (b) $\lambda_{x_i} = \lambda_{x_j} - (d_T(i) - 1)\lambda_{z_{lm}} \forall i \in \bar{L}_T, j \in V \setminus \bar{L}_T, lm \in E_T$. If $|E_T| = 1$, statement (a) is trivial, so we may assume that one of i, j belongs to \bar{L}_T and, w.l.o.g., we assume $i \in \bar{L}_T$. Consider T as rooted in i . Let T_1, \dots, T_q be the subtrees obtained from $T \setminus \{i\}$, and let T_u be an arbitrary subtree with $1 \leq u \leq q$. As $i \in L_T$, we know $q \geq 2$. Since $|V| \geq k + 1 + b(T)$, for each $1 \leq r \leq q, |V| \geq k + 1 + |T_r|$. This allows us to build a feasible solution $w^1 = (x, z) \in F$, in which $w_{x_t}^1 = 1$ for all $t \in V(M)$, $w_{z_{ts}}^1 = 1$ for all $ts \in M$, $w_{x_i}^1 = 0, w_{x_d}^1 = 1$ for all $d \in \bigcup_{1 \leq r \leq q, r \neq u} V(T_r)$, $w_{z_{de}}^1 = 1$ for all $de \in \bigcup_{1 \leq r \leq q, r \neq u} E(T_r)$, and the remaining $|T_u|$ vertices are selected from $V \setminus (V(M) \cup V_T)$. Let j_r be the root of T_r , for $r = 1, \dots, q$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_v}^2 = 0$ for some vertex $v \in V \setminus (V(M) \cup V_T)$, $w_{x_t}^2 = 1$ for some leaf $t \in L_T \cap T_u$. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, implying $\lambda_{x_v} = \lambda_{x_t}$. Since i, v, u , and t are arbitrary,

$$\lambda_{x_v} = \lambda_{x_t} \text{ for every } v \in V \setminus (V(M) \cup V_T), t \in L_T. \tag{A.1}$$

Now, let $w^3 \in F$ be obtained from w^1 by setting $w_{x_v}^3 = 0$ for some vertex $v \in V \setminus (V(M) \cup V_T)$, $w_{x_i}^3 = 1, w_{z_{ijr}}^3 = 1$ for $r = 1, \dots, q, r \neq u$, and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^3$, implying

$$\lambda_{x_v} = \lambda_{x_i} + \sum_{1 \leq r \leq q, r \neq u} \lambda_{z_{ijr}}. \tag{A.2}$$

Using the same construction as before but considering a different $T_l (l \neq u)$, we obtain $\lambda_{x_v} = \lambda_{x_i} + \sum_{1 \leq r \leq q, r \neq l} \lambda_{z_{ijr}}$, thus concluding $\lambda_{z_{ij}} = \lambda_{z_{ju}}$ for every $j_u, j_l \in N_T(i)$. Pivoting on each internal vertex in the path from ij to lm in T , we obtain $\lambda_{z_{ij}} = \lambda_{z_{lm}}$ and thus prove Part (a).

Notice that (A.2) and (A.1), together with Claim 2 and Part (a) of this claim, imply $\lambda_{x_i} = \lambda_{x_j} - (d_T(i) - 1)\lambda_{z_{lm}}$, for $i \in \bar{L}_T, j \in V \setminus \bar{L}_T, lm \in E_T$.

Claim 4. $\lambda_{z_{ij}} = \lambda_{z_{lq}} \forall ij \in M, lq \in E_T$. Consider the feasible solution $w^1 = (x, z) \in F, w_{x_i}^1 = 0$ for the vertex $i, w_{x_m}^1 = 1$ for the remaining vertices $m \in V(M), w_{z_{mt}}^1 = 1$ for $mt \in M, m, t \neq i, w_{x_t}^1 = 1$ for all $t \in V_T$ and $w_{z_{tu}}^1 = 1$ for all $tu \in E_T$. Let $w^2 \in F$ be obtained from w^1 by setting $w_{x_l}^2 = 0$ for some leaf $l \in L_T, w_{z_{lq}}^2 = 0$ for the corresponding edge $lq \in E_T, w_{x_i}^2 = 1, w_{z_{ij}}^2 = 1$ for the corresponding edge $ij \in M$ and keeping the remaining variables unchanged. Since both points belong to F , we have $\lambda^T w^1 = \lambda^T w^2$, and together with Claims 2 and 3 this implies that $\lambda_{z_{ij}} = \lambda_{z_{lq}}$.

Defining $\alpha := \lambda_{z_{ij}}$ for $ij \in E_T$ and $\beta := \lambda_{x_i}$ for $i \in V \setminus \bar{L}_T$, we obtain that $\lambda_{z_{ij}} = \alpha$ for every $ij \in M$ using Claim 4, and that $\lambda_{x_i} = \beta - (d_T(i) - 1)\alpha$ for every $i \in \bar{L}_T$ using Claim 3. Notice that $\lambda_{z_{jl}} = 0$ for every $jl \in E \setminus (M \cup E_T)$ by Claim 1. We conclude that λ is a linear combination of the coefficient vector of the matching inequality (14) and the coefficient vector of the model constraint (1b). A calculation similar to (3) shows that λ_0 can be written as the same combination of the independent terms of (14) and (1b), hence (14) induces a facet of $P(G, k)$. \square

References

[1] J.M. Abello, P. Pardalos, M.G.C. Resende, On maximum clique problems in very large graphs, in: External Memory Algorithms, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 50, American Mathematical Society, Boston, MA, USA, 1999, pp. 119–130.
 [2] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Reviews of Modern Physics 74 (2002) 47–97.
 [3] Y. Asahiro, R. Hassin, K. Iwama, Complexity of finding dense subgraphs, Discrete Applied Mathematics 121 (2002) 15–26.
 [4] Y. Asahiro, K. Iwama, H. Tamaki, T. Tokuyama, Greedily finding dense subgraphs, Journal of Algorithms 34 (2000) 203–221.
 [5] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512.
 [6] A. Billionnet, Different formulations for solving the heaviest k -subgraph problem, Information Systems and Operational Research 43 (3) (2005) 171–186.
 [7] F. Bonomo, J. Marengo, D. Saban, N.E. Stier-Moses, A polyhedral study of the maximum edge subgraph problem, in: Proceedings of the V Latin-American Algorithms, Graphs and Optimization Symposium, LAGOS'09, Gramado, Brazil, in: Electronic Notes in Discrete Mathematics, vol. 35, 2009, pp. 197–202.
 [8] D. Corneil, Y. Perl, Clustering and domination in perfect graphs, Discrete Applied Mathematics 9 (1984) 27–39.
 [9] J. Edmonds, Maximum matching and a polyhedron with $(0, 1)$ -vertices, Journal of Research of the National Bureau of Standards 69 (1965) 125–130.
 [10] J. Edmonds, Paths, trees, and flowers, Canadian Journal of Mathematics 17 (1965) 449–467.
 [11] P. Erdős, A. Rényi, On random graphs, Publicationes Mathematicae 6 (1959) 290–297.
 [12] U. Feige, G. Kortsarz, D. Peleg, The dense k -subgraph problem, Algorithmica 29 (3) (2001) 410–421.
 [13] M. Fischetti, H. Hamacher, K. Jørnsten, F. Maffioli, Weighted k -cardinality trees: complexity and polyhedral structure, Networks 24 (1994) 11–21.
 [14] A.A. Hagberg, D.A. Schult, P.J. Swart, Exploring network structure, dynamics, and function using NetworkX, in: Proceedings of the 7th Python in Science Conference, SciPy'08, Pasadena, CA, USA, August 2008, pp. 11–15.
 [15] Q. Han, Y. Ye, J. Zhang, Approximation of dense- k -subgraph, Technical Report, Department of Management Sciences, Henry B. Tippie College of Business, the University of Iowa, 2000.
 [16] M. Jünger, S. Thienel, The ABACUS system for branch-and-cut-and-price algorithms in integer programming and combinatorial optimization, Software: Practice and Experience 30 (2000) 1325–1352.
 [17] E. Johnson, A. Mehrotra, G. Nemhauser, Min-cut clustering, Mathematical Programming 62 (1993) 133–151.
 [18] D.S. Johnson, M.A. Trick, Cliques, Coloring, and Satisfiability: Second Dimacs Implementation Challenge, October 11–13, 1993, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26, American Mathematical Society, Providence, RI, 1996.
 [19] M. Liazi, I. Milis, F. Pascual, V. Zissimopoulos, The densest k -subgraph problem on clique graphs, Journal of Combinatorial Optimization 14 (4) (2007) 465–474.

- [20] M. Liazi, I. Milis, V. Zissimopoulos, A constant approximation algorithm for the densest k -subgraph problem on chordal graphs, *Information Processing Letters* 108 (1) (2008) 29–32.
- [21] A. Mehrotra, Cardinality constrained Boolean quadratic polytope, *Discrete Applied Mathematics* 79 (1997) 137–154.
- [22] M. Padberg, The boolean quadric polytope: some characteristics, facets and relatives, *Mathematical Programming* 45 (1) (1989) 139–172.
- [23] F. Roupin, A. Billionnet, A deterministic approximation algorithm for the densest k -subgraph problem, *International Journal of Operational Research* 3 (3) (2008) 301–314.
- [24] D. Saban, F. Bonomo, N.E. Stier-Moses, Analysis and models of bilateral investment treaties using a social networks approach, *Physica A: Statistical Mechanics and its Applications* 389 (2010) 3661–3673.
- [25] H. Sherali, J. Cole Smith, A polyhedral study of the generalized vertex packing problem, *Mathematical Programming* 107 (3) (2006) 367–390.
- [26] S. Wasserman, K. Faust, *Social Network Analysis: Methods and Applications*, in: *Structural Analysis in the Social Sciences*, vol. 8, Cambridge University Press, 1994.