




Declawing a graph: polyhedra and Branch-and-Cut algorithms

Felipe C. Fragoso¹ · Gilberto F. de Sousa Filho¹  · Fábio Protti²

Accepted: 7 April 2021 / Published online: 19 April 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The complete bipartite graph $K_{1,3}$ is called a *claw*. A graph is said to be *claw-free* if it contains no induced subgraph isomorphic to a claw. Given a graph G , the NP-hard *Graph Declawing Problem* (GDP) consists of finding a minimum set $S \subseteq V(G)$ such that $G - S$ is claw-free. This work develops a polyhedral study of the GDP polytope, expliciting its full dimensionality, proposing and testing five families of facets: trivial inequalities, claw inequalities, star inequalities, lantern inequalities, and binary star inequalities. In total, four Branch-and-Cut algorithms with separation heuristics have been developed to test the computational benefits of each proposed family on random graph instances and random interval graph instances. Our results show that the model that uses a separation heuristics proposed for star inequalities achieves better results on both set of instances in almost all cases.

Keywords Graph declawing problem · Claw-free graphs · Branch-and-cut · Polyhedral combinatorics

1 Introduction

A graph G is said to be *claw-free* if it contains no induced subgraph isomorphic to the complete bipartite graph $K_{1,3}$, also called *claw*. The *Graph Declawing Problem* (GDP, for short) consists of finding a minimum set $S \subseteq V(G)$ such that $G - S$ is claw-free. Lewis and Yannakakis (1980) prove a general result showing that any vertex deletion

✉ Gilberto F. de Sousa Filho
gilberto@ci.ufpb.br

Felipe C. Fragoso
felipefragoso@ppgi.ci.ufpb.br

Fábio Protti
fabio@ic.uff.br

¹ Centro de Informática, Universidade Federal da Paraíba (UFPB), João Pessoa, PB, Brazil

² Instituto de Computação, Universidade Federal Fluminense (UFF), Niterói, RJ, Brazil

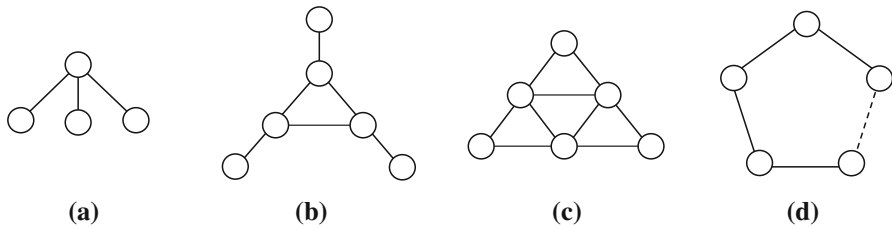


Fig. 1 Forbidden subgraphs for unit interval graphs: **a** claw, **b** net, **c** tent, **d** C_k , $k \geq 4$

problem on undirected/directed graphs for a *nontrivial* and *hereditary* property Π is NP-Hard. Π is considered to be nontrivial if it is true for infinitely many graphs and false for infinitely many graphs, and hereditary if, for any graph H satisfying Π , every vertex-induced subgraph of H also satisfies Π . Clearly, being claw-free is a hereditary and nontrivial property; therefore, the GDP is NP-Hard.

The GDP is related to the resolution of the *Unit Interval Vertex Deletion Problem*, which consists of converting a given graph into a unit interval graph by removing a minimum subset of vertices such that the resulting graph is free of the following forbidden induced subgraphs: claws, *nets*, *tents*, and induced cycles C_k ($k \geq 4$) (see Fig. 1). Studies on the parameterized version of this problem, including kernelization, are developed by van Bevern et al. (2010), van't Hof and Villanger (2013), Fomin et al. (2013), and Ke et al. (2018).

The GDP is also related with the problem of converting an *interval graph* into a *unit interval graph*. An interval graph is the intersection graph of a family of intervals. Formally speaking, G is an interval graph if its vertices can be associated with intervals on the real line such that $uv \in E(G)$ if and only if $I_u \cap I_v \neq \emptyset$, where I_u and I_v are the intervals associated with u and v . A unit interval graph is an interval graph where the intervals can be chosen so that all of them have the same length (or, equivalently, length one). Figure 2 shows a transformation from an interval graph to a unit interval graph.

Unit interval graphs coincide with *proper interval graphs* (interval graphs where the intervals can be chosen so that no interval properly contains another) and *indifference graphs* (defined below). See Bogart and West (1999) for a proof that G is a unit interval graph if and only if G is a proper interval graph, and Roberts (1979) for a proof that G is a unit interval graph if and only if G is an indifference graph.

Indifference graphs are defined in the context of Social Sciences, as discussed in the work by Roberts (1979). A graph G is an indifference graph if there is a function $f : V(G) \rightarrow \mathbb{R}$ such that

$$(u, v) \in E(G) \iff |f(u) - f(v)| \leq \epsilon,$$

where ϵ is a positive number that measures *closeness*. Informally, this means that u and v are indistinguishable items if and only if, according to f , there is a difference at most ϵ between their values. But since item values are in general empirically obtained, there is the possibility that some presumably close items are distinguishable. Therefore, one may be interested in removing the fewer items possible to generate a real indifference

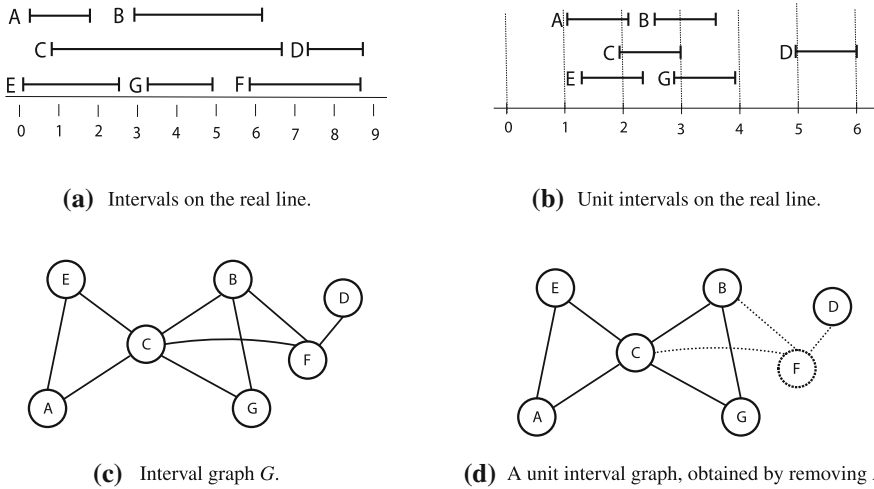


Fig. 2 Converting an interval graph **c** associated with **a** into a unit interval graph **d** associated with **b** by removing *F* from **c**

graph. Since indifference graphs are precisely the claw-free interval graphs (Roberts 1969), this problem is precisely the GDP applied to a given interval graph as input, as the process exemplified in Fig. 2. Fishburn (1985) poses the problem of converting an *interval order* into an *indifference order* by the removal of some elements; again, in terms of graphs, this problem is precisely the problem of optimally declawing an interval graph. We remark that the complexity of this problem is still an open question.

Williams et al. (2015) summarize complexity results for the recognition of 4-node induced subgraphs; in particular, claws have an $O(n^{3.257})$ recognition time. Bonomo-Braberman et al. (2020) prove that the GDP is hard to approximate within a constant factor better than 2 assuming the *Unique Games Conjecture*; moreover, they show that the weighted problem associated with the GDP can be solved in polynomial time for graphs with bounded tree-width and for block graphs. Elimination of claws and diamonds via edge deletion was studied by Cygan et al. (2016). Aravind et al. (2017) present some general results on edge elimination problems, and present the first polynomial kernel for elimination of claws via edge deletion under the assumption of a K_t -free input graph.

Properties of claw-free graphs have been deeply investigated by the research community (see the survey by Faudree et al. (1997)). For example, Minty (1980) presents a polynomial-time algorithm for the problem of finding an independent set with maximum weight in a claw-free graph. Hsu and Nemhauser (1982) presents a polynomial-time algorithm for the Minimum Weighted Clique Cover Problem in claw-free perfect graphs. Zhang (1988) proves Bondy’s Conjecture for claw-free graphs. Broersma et al. (2011) present two algorithms that solve the Longest Cycle Problem in polynomial time in claw-free graphs. Hermelin et al. (2014) show that induced subgraph isomorphism for fixed graphs is fixed parameter tractable on claw-free graphs. Hermelin et al. (2019) show that dominating set is fixed parameter tractable

on claw-free graphs. Martin et al. (2020) prove that the Disconnected Cut Problem is polynomially solvable for claw-free graphs.

Another application of claw-free graphs appear in resource constrained scheduling. Suppose a set of n jobs J_1, J_2, \dots, J_n in a distributed computing environment, where each J_i needs exclusive access to a set R_i of resources to execute. Let G be the conflict graph of J_1, J_2, \dots, J_n , where each vertex v_i of G is associated with job J_i , and two vertices v_i and v_j , $i \neq j$, are linked by an edge if and only if $R_i \cap R_j \neq \emptyset$. In fact, G can be viewed as the intersection graph of a hypergraph, whose hyperedges are R_1, R_2, \dots, R_n . If each job needs access to at most k resources, i.e., $|R_i| \leq k$ for every $i = 1, \dots, n$, then the conflict graph G is $(k+1)$ -claw free, where a $(k+1)$ -claw in G is an induced subgraph of G isomorphic to a star $K_{1,k+1}$. (In this terminology, a 3-claw is a claw.) See details in (Halldórsson et al. 2003). Note that an independent set in G corresponds to a set of jobs that can execute simultaneously. Finding a maximum independent set in a $(k+1)$ -claw free graph is NP-hard for $k \geq 3$ and solvable in polynomial time for $k = 2$. Thus, if $k \geq 3$, one can solve the GDP for the conflict graph, obtaining a claw-free conflict graph G' , and then find in polynomial time a set of jobs with no conflict.

Up to the author's knowledge, there is no work in the literature presenting a polyhedral study of the GDP or a method to solve it via integer programming. Therefore, we believe that the studies developed in this work are fully justified.

The remainder of this work is organized as follows. Section 2 defines the GDP polytope and some facet-defining inequalities. Section 3 presents more facet-defining inequalities that require deeper analyses, along with a more precise description of the GDP polytope. Section 4 describes separation heuristics for support graphs needed to generate the inequalities. Section 5 shows computational results for random graphs and interval graph instances. Section 6 contains our conclusions.

2 The graph declawing problem polytope

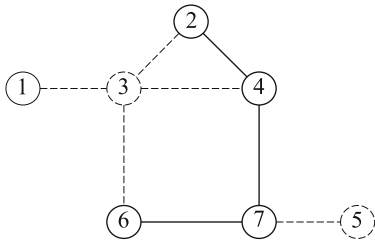
From now on, G denotes a graph with $|V(G)| = \{1, \dots, n\}$. Let P_m be the convex hull of m incidence vectors χ^H that represent claw-free subgraphs H of a graph G , that is:

$$P_m = \text{conv} \left\{ \chi^H \in \{0, 1\}^n \mid H \text{ is a claw-free subgraph of } G \right\}.$$

Say that P_m is the *GDP polytope* of G . Figure 3 shows a claw-free subgraph H induced by $\{1, 2, 4, 6, 7\}$ and its incidence vector.

The GDP can be described as an optimization problem that aims at finding an incidence vector $x^T = (x_1, x_2, \dots, x_n)$ with minimum number of zeros (removed vertices):

$$\begin{aligned} \min & \sum_{i=1}^n (1 - x_i) \\ \text{s.t. } & x \in P_m \end{aligned}$$



$$\chi^H = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

(a) Claw-free subgraph H

(b) Incidence vector $\chi^H \in \{0, 1\}^7$

Fig. 3 a A claw-free subgraph $H = G[\{1, 2, 4, 6, 7\}]$ and b its incidence vector

In order to apply binary integer programming techniques for the GDP, we focus on the characterization of P_m , which is n -dimensional and contains the null and all the unit vectors, summing up to $n + 1$ affinely independent vectors. This means that P_m is full dimensional. The full dimensionality of a polytope implies that every facet of it is uniquely defined by an inequality multiplied by some scalar α . This fact will be used in the next subsections for proofs of facets. Another useful fact is that a facet of an n -dimensional polytope must have n affinely independent points.

2.1 Trivial inequalities

Theorem 1 *The following statements are true:*

1. *Trivial inequalities $0 \leq \chi^H$ and $\chi^H \leq 1$ are valid for P_m .*
2. *Inequalities $\chi^H \geq 0$ induce facets of P_m .*
3. *Inequalities $\chi^H \leq 1$ induce facets of P_m .*

Proof 1. Since all the coordinates of the m incidence vectors of claw-free subgraphs are binary, it is clear that the trivial inequalities are valid for P_m .

2. Let x_v represent the coordinate $v \in \{1, \dots, n\}$ of a vector x . Then $\chi_v^H = 0$ is satisfied by the null vector and all the unit vectors x with $x_u = 1, u \neq v$. These n vectors belong to P_m and are affinely independent.
3. $\chi_v^H = 1$ is satisfied by the unit vector x with $x_v = 1$ and by the vectors w such that $w_v = 1, w_u = 1$ for some $u \neq v$, and $w_t = 0$ for $t \notin \{u, v\}$. These n vectors belong to P_m and are affinely independent.

□

2.2 Claw inequalities

Say that an induced subgraph of G isomorphic to $K_{1,3}$ is a *claw subgraph*. For simplicity, we denote by $abcd$ a claw subgraph with central vertex a (the vertex with degree three in the claw subgraph). In addition, for a subset S of vertices, sometimes we simply write χ^S to mean the incidence vector of the induced subgraph $G[S]$.

Theorem 2 *Let*

$$x_a + x_b + x_c + x_d \leq 3 \tag{1}$$

be the claw inequality associated with a claw subgraph $abcd$. Then:

1. The claw inequality is valid for P_m .
2. The claw inequality induces a facet of P_m .

Proof 1. Immediate. A claw inequality forces at least one of x_a, x_b, x_c, x_d to be zero.
 2. Let w be a vector such that $w_i = 1$ for $i \in \{a, b, c, d\}$ and $w_i = 0$ for $i \notin \{a, b, c, d\}$. Let $u^T x \leq u_0$ be a facet-inducing inequality for the GDP polytope P_m such that $F_a = \{\chi^H \in P_m \mid w^T \chi^H = 3\} \subseteq F_b = \{\chi^H \in P_m \mid u^T \chi^H = u_0\}$. Clearly, $F_a \neq P_m$ and $F_a \neq \emptyset$. Thus, a proof that $u = \alpha w$ for some $\alpha \in \mathbb{R}$ shows that F_a defines a facet of P_m .

For a claw subgraph $abcd$, the incidence vectors $\chi^{\{a,b,c\}}, \chi^{\{a,b,d\}}, \chi^{\{b,c,d\}}$, and $\chi^{\{a,c,d\}}$ are in $F_a \subseteq F_b$. Then, the following equalities hold: $0 = u_0 - u_0 = u^T \chi^{\{a,b,c\}} - u^T \chi^{\{a,b,d\}} = u_c - u_d; 0 = u_0 - u_0 = u^T \chi^{\{a,b,c\}} - u^T \chi^{\{a,c,d\}} = u_b - u_d; 0 = u_0 - u_0 = u^T \chi^{\{a,b,c\}} - u^T \chi^{\{b,c,d\}} = u_a - u_d$. This implies $u_a = u_b = u_c = u_d = \alpha$.

For a node $v \notin \{a, b, c, d\}$, if its neighborhood satisfies $|N(v) \cap \{a, b, d\}| \leq 2$ then $\chi^{\{a,b,d,v\}} \in F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{a,b,d,v\}} - u^T \chi^{\{a,b,d\}} = u_v$; That means $u_v = 0$.

Finally, for a vertex $v \notin \{a, b, c, d\}$ such that $|N(v) \cap \{a, b, d\}| \geq 3$, $\chi^{\{a,b,c,v\}} \in F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{a,b,c,v\}} - u^T \chi^{\{a,b,c\}} = u_v$. □

3 Facets derived from star, lantern, and binary star graphs

3.1 Star inequalities

Definition 1 A star graph S_k is a complete bipartite graph $K_{1,k}$, for $k \geq 3$. S_k has a central vertex c whose neighborhood is an independent set I_k of size k . Possible topologies for S_k are illustrated in Fig. 4.

A star subgraph is an induced subgraph of G isomorphic to a star graph.

Theorem 3 Let S_k be a star subgraph, and consider the corresponding star inequality:

$$\sum_{v \in I_k} x_v + (k - 2)x_c \leq k. \tag{2}$$

Then:

1. The star inequality is valid for P_m .
2. The star inequality induces a facet of P_m .

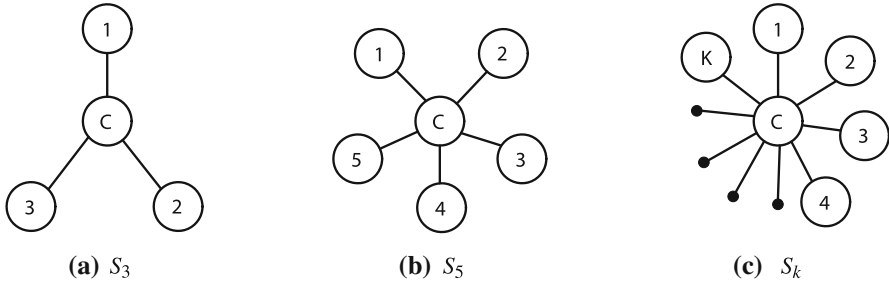


Fig. 4 Star graphs S_3 , S_5 , and S_k

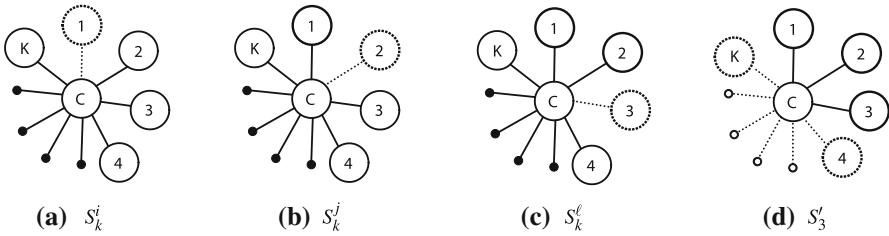


Fig. 5 Subgraphs S_k^i , S_k^j , S_k^l , and S'_3 of S_k

Proof 1. The case $k = 3$ corresponds to the claw inequality (see Theorem 2). For $k \geq 4$, the proof is done by induction. Let $S_k^v = S_k - v$ and $S'_3 = S_k[\{i, j, \ell, c\}]$, where i, j, ℓ are distinct vertices in I_k . Note that S_k^i , S_k^j , S_k^l , and S'_3 are subgraphs of S_k . See Fig. 5, where $i = 1, j = 2, \ell = 3$.

By induction, the inequalities associated with S_k^i , S_k^j , S_k^l , and S'_3 are valid for P_m , and summing them up leads to the valid inequality

$$\sum_{v \in I_k} 3x_v + (3k - 8)x_c \leq 3k.$$

Adding $2x_c \leq 2$ to the above inequality gives

$$\sum_{v \in I_k} 3x_v + (3k - 6)x_c \leq 3k + 2,$$

and, therefore,

$$\sum_{v \in I_k} x_v + (k - 2)x_c \leq k + \left\lfloor \frac{2}{3} \right\rfloor = k$$

is valid for P_m .

2. Let w be a vector such that $w_c = k - 2, w_i = 1$ for $i \in I_k$, and $w_v = 0$ for $v \notin I_k \cup \{c\}$. Let $u^T x \leq u_0$ be a facet-inducing inequality for the GDP polytope

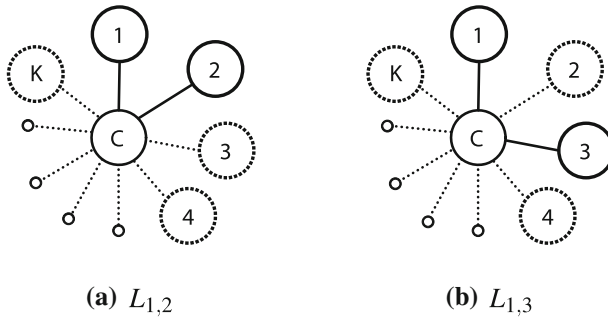


Fig. 6 Subgraphs $L_{1,2}$ and $L_{1,3}$ of S_k

P_m such that $F_a = \{\chi^H \in P_m \mid w^T \chi^H = k\} \subseteq F_b = \{\chi^H \in P_m \mid u^T \chi^H = u_0\}$. Clearly, $F_a \neq P_m$ and $F_a \neq \emptyset$. Thus, a proof that $u = \alpha w$ for some $\alpha \in \mathbb{R}$ shows that F_a defines a facet of P_m .

Let $L_{i,j}$ be a claw-free subgraph of S_k such that $L_{i,j} = S_k[\{i, j, c\}]$, where $\{i, j\} \subset I_k$ and c is the center of S_k . Figure 6 shows two examples of such subgraphs.

The incidence vectors χ^{I_k} , $\chi^{L_{i,j}}$, and $\chi^{L_{j,k}}$ are in $F_a \subseteq F_b$. Then, the following equalities hold: $0 = u_0 - u_0 = u^T \chi^{L_{i,j}} - u^T \chi^{L_{j,k}} = u_i - u_k$. Therefore, $u_i = u_k = \alpha$. By applying this process to all $v \in I_k$ we get $u_v = \alpha$. Also, $0 = u_0 - u_0 = u^T \chi^{I_k} - u^T \chi^{L_{i,j}} = (\sum_{v \in I_k \setminus \{i,j\}} u_v) - u_c$, and this implies $u_c = \alpha(k - 2)$.

For a node $v \notin I_k \cup \{c\}$, if its neighborhood satisfies $|N(v) \cap I_k| < 3$ then $\chi^{I_k \cup \{v\}} \in F_a \subseteq F_b$. Thus, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{I_k \cup \{v\}} - u^T \chi^{I_k} = u_v$.

Finally, for a node $v \notin I_k \cup \{c\}$ such that $|N(v) \cap I_k| \geq 3$, the incidence vector $\chi^{V(L_{i,j}) \cup \{v\}}$, where $\{i, j\} \subset N(v) \cap I_k$, is in $F_a \subseteq F_b$. Then, $0 = u_0 - u_0 = u^T \chi^{V(L_{i,j}) \cup \{v\}} - u^T \chi^{L_{i,j}} = u_v$. \square

3.2 Lantern inequalities

Definition 2 Let S_k and $S_\ell, k > \ell \geq 3$, be two star graphs such that S_ℓ has central node c_1 and independent set I_ℓ, S_k has central node c_2 and independent set I_k , and $I_\ell \subset I_k$. A lantern graph $L_{\ell k}$ is the union of S_ℓ and S_k . Figure 7 shows some topologies of lantern graphs.

A lantern subgraph is an induced subgraph isomorphic to a lantern graph.

Theorem 4 Let $L_{\ell k}$ be a lantern subgraph, and consider the corresponding lantern inequality:

$$\sum_{v \in I_k} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq k. \tag{3}$$

Then:

1. The lantern inequality is valid for P_m .

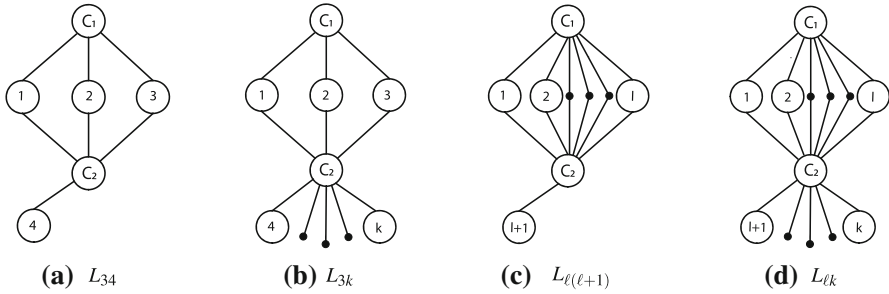


Fig. 7 Lantern graphs L_{34} , L_{3k} , $L_{\ell(\ell+1)}$, and $L_{\ell k}$

2. The lantern inequality induces a facet of P_m , except when $k = \ell + 1$ and there is a vertex $v \notin V(L_{\ell k})$ such that $N(v) \cap I_k = I_k$ and $N(v) \cap \{c_1, c_2\} = \emptyset$.

Proof 1. The validity of the lantern inequality for $\ell = 3$ and $k = 4$ can be checked by adding the star inequalities associated with the subgraphs $S'_3 = L_{34}[\{1, 2, 3, c_1\}]$ and $S'_4 = L_{34}[\{1, 2, 3, 4, c_2\}]$ and the inequality $x_{c_1} + x_4 \leq 2$, obtaining the following valid inequality:

$$\sum_{v \in I_4} x_v + x_{c_1} + x_{c_2} \leq 4 + \left\lfloor \frac{1}{2} \right\rfloor = 4.$$

Now, consider $L_{\ell k}^v = L_{\ell k} - v$. The case $\ell = 3, k = 5$ can be checked by adding the inequalities associated with subgraphs L_{35}^4, L_{35}^5 (where $\{4, 5\} \subset I_5$), $S'_3 = L_{35}[\{1, 4, 5, c_2\}]$, $S'_5 = L_{35}[\{1, 2, 3, 4, 5, c_2\}]$ and the valid inequality $x_{c_1} - x_1 \leq 1$, obtaining the following inequality:

$$\sum_{v \in I_5} x_v + x_{c_1} + 2x_{c_2} \leq 5 + \left\lfloor \frac{2}{3} \right\rfloor = 5.$$

For the case $\ell = 3$ and $k > 5$, the proof is done by induction. Note that $L_{3k}^i, L_{3k}^j, L_{3k}^h$, and $S'_3 = L_{3k}[\{i, j, h, c_2\}]$, for distinct $i, j, h \in I_k \setminus I_3$, are subgraphs of L_{3k} . Figure 8 shows subgraphs $L_{3k}^i, L_{3k}^j, L_{3k}^h$, and S'_3 for $i = 4, j = k - 1, h = k$.

By induction, the inequalities associated with $L_{3k}^i, L_{3k}^j, L_{3k}^h$, and S'_3 are valid for P_m , and summing them up leads to the valid inequality

$$3 \sum_{v \in I_k} x_v + 3(3 - 2)x_{c_1} + (3k - 11)x_{c_2} \leq 3k.$$

Adding $2x_{c_2} \leq 2$, we get that

$$\sum_{v \in I_k} x_v + (3 - 2)x_{c_1} + (k - 3)x_{c_2} \leq k + \left\lfloor \frac{2}{3} \right\rfloor = k$$

is valid for P_m .

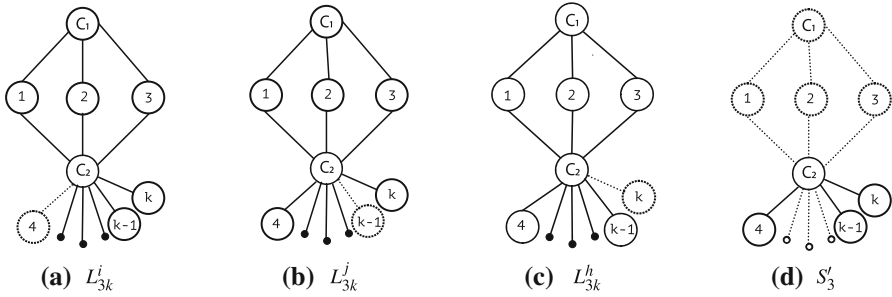


Fig. 8 Subgraphs of L_{3k} , $k > 5$

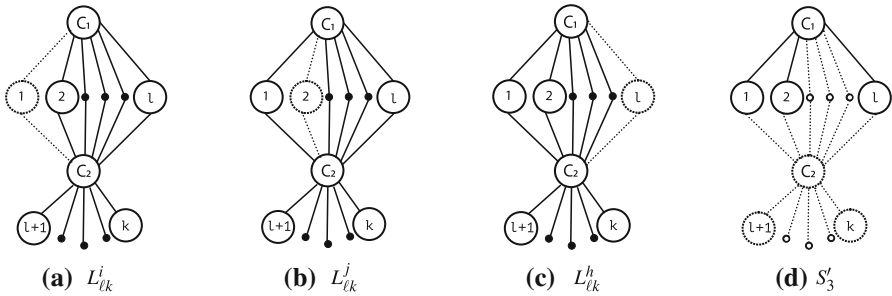


Fig. 9 Subgraphs of $L_{\ell k}$ for $l > 3$, $k \geq l + 1$

For the case $\ell > 3$ and $k \geq \ell + 1$, note that $L_{\ell k}^i, L_{\ell k}^j, L_{\ell k}^h$, and $S'_3 = L_{\ell k}[\{i, j, h, c_1\}]$, for distinct $i, j, h \in I_\ell$, are subgraphs of $L_{\ell k}$. Figure 9 shows subgraphs $L_{\ell k}^i, L_{\ell k}^j, L_{\ell k}^h$, and S'_3 for $i = 1, j = 2, h = \ell$.

By induction, the inequalities associated with $L_{\ell k}^i, L_{\ell k}^j, L_{\ell k}^h$, and S'_3 are valid for P_m , and summing them up leads to

$$3 \sum_{v \in I_k} x_v + (3\ell - 8)x_{c_1} + 3(k - \ell)x_{c_2} \leq 3k.$$

Adding $2x_{c_1} \leq 2$ we get that

$$\sum_{v \in I_k} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq k \left\lfloor \frac{2}{3} \right\rfloor = k$$

is valid for P_m .

- Let w be a vector such that $w_{c_1} = \ell - 2, w_{c_2} = k - \ell, w_i = 1$ for $i \in I_k$, and $w_v = 0$ for $v \notin V(L_{\ell k})$. Let $u^T x \leq u_0$ be a facet-inducing inequality for the GDP polytope P_m such that $F_a = \{\chi^H \in P_m \mid w^T \chi^H = k\} \subseteq F_b = \{\chi^H \in P_m \mid u^T \chi^H = u_0\}$. Clearly, $F_a \neq P_m$ and $F_a \neq \emptyset$. Thus, a proof that $u = \alpha w$ for some $\alpha \in \mathbb{R}$ shows that $w^T \chi^H = k$ defines a facet of P_m .

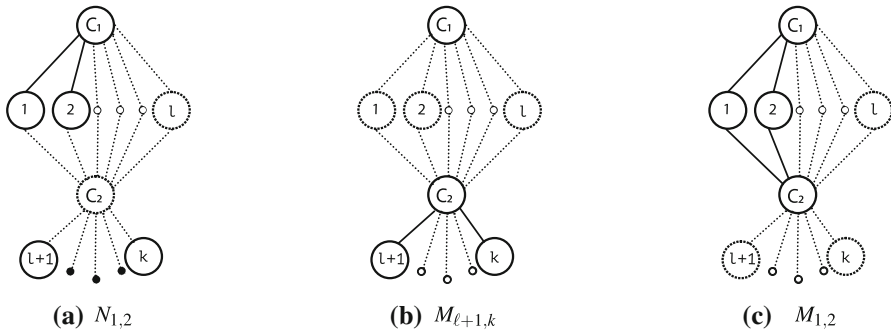


Fig. 10 Subgraphs of $L_{\ell k}$

Let $N_{i,j}$ and $M_{i,j}$ be claw-free subgraphs of $L_{\ell k}$ such that $N_{i,j} = L_{\ell k}[(I_k \setminus I_\ell) \cup \{i, j, c_1\}]$ and $M_{i,j} = L_{\ell k}[\{i, j, c_1, c_2\}]$, where $i, j \in I_k, i \neq j$. Figure 10 shows examples of subgraphs $N_{i,j}$ and $M_{i,j}$.

Then:

- a) The incidence vectors $\chi^{M_{i,j}}$ and $\chi^{M_{j,h}}$ with $\{i, j, h\} \subset I_k$ are in $F_a \subseteq F_b$. Thus, $0 = u_0 - u_0 = u^T \chi^{M_{i,j}} - u^T \chi^{M_{j,h}} = u_i - u_h$. This implies $u_i = u_h = \alpha$ for all choices of $\{i, j, h\} \subset I_k$.
- b) The incidence vectors χ^{I_k} and $\chi^{N_{i,j}}$ with $\{i, j\} \subset I_\ell$ are in $F_a \subseteq F_b$. Thus, $0 = u_0 - u_0 = u^T \chi^{I_k} - u^T \chi^{N_{i,j}} = (\sum_{v \in I_\ell \setminus \{i,j\}} u_v) - u_{c_1}$. This implies $u_{c_1} = \alpha(\ell - 2)$.
- c) The incidence vectors χ^{I_k} and $\chi^{M_{i,j}}$ with $\{i, j\} \subset I_k$ are in $F_a \subseteq F_b$. Thus $0 = u_0 - u_0 = u^T \chi^{I_k} - u^T \chi^{M_{i,j}} = (\sum_{v \in I_k \setminus \{i,j\}} u_v) - u_{c_1} - u_{c_2}$. This implies $u_{c_2} = \alpha(k - 2) - \alpha(\ell - 2) = \alpha(k - \ell)$.

For a node $v \notin V(L_{\ell k})$ such that $|N(v) \cap I_k| < 3, \chi^{\{v\} \cup I_k}$ is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup I_k} - u^T \chi^{I_k} = u_v$.

For a node $v \notin V(L_{\ell k})$ such that $|N(v) \cap I_k| > 3$, we have that:

- a) If $k = \ell + 1$ and $k \notin N(v), \chi^{\{v\} \cup V(N_{i,j})}$ with $i \in I_\ell \cap N(v)$ and $j \in I_\ell$ is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(N_{i,j})} - u^T \chi^{N_{i,j}} = u_v$.
- b) If $k = \ell + 1, k \in N(v)$, and $N(v) \cap I_k \neq I_k, \chi^{\{v\} \cup V(N_{i,j})}$ with $i \in I_\ell \cap N(v)$ and $j \notin I_\ell \cap N(v)$ is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(N_{i,j})} - u^T \chi^{N_{i,j}} = u_v$.
- c) If $k = \ell + 1, N(v) \cap I_k = I_k$, and $c_1 \in N(v)$ or $c_2 \in N(v), \chi^{\{v\} \cup V(M_{i,j})}$ with $i \in I_\ell$ and $j = k$ is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(M_{i,j})} - u^T \chi^{M_{i,j}} = u_v$.
- d) If $k > \ell + 1$ and there is no vertex $i \in (I_k \setminus I_\ell) \cap N(v), \chi^{\{v\} \cup V(N_{i,j})}$ with $i \in I_\ell \cap N(v)$ and $j \in I_\ell$ is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(N_{i,j})} - u^T \chi^{N_{i,j}} = u_v$.
- e) If $k > \ell + 1, c_1 \notin N(v)$, and there is a vertex $i \in (I_k \setminus I_\ell) \cap N(v)$, the incidence vector $\chi^{\{v\} \cup V(M_{i,j})}$ is in $F_a \subseteq F_b$, for $j \in I_k \setminus I_\ell$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(M_{i,j})} - u^T \chi^{M_{i,j}} = u_v$.
- f) If $k > \ell + 1, c_1 \in N(v)$, and there is a vertex $i \in (I_k \setminus I_\ell) \cap N(v)$, the incidence vector $\chi^{\{v\} \cup V(M_{i,j})}$ is in $F_a \subseteq F_b$, for $j \in I_\ell$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(M_{i,j})} - u^T \chi^{M_{i,j}} = u_v$.

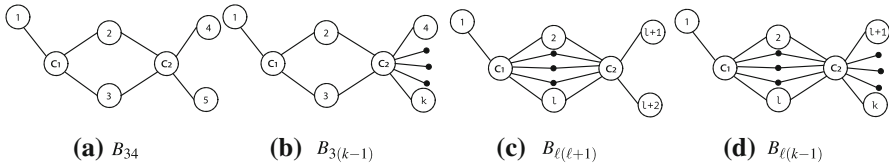


Fig. 11 Examples of binary stars

Note that the only case not covered by the above items (a)-(f) is $k = \ell + 1$, $N(v) \cap I_k = I_k$, and $N(v) \cap \{c_1, c_2\} = \emptyset$. \square

3.3 Binary star inequalities

Definition 3 Let S_ℓ and S_k be two star graphs, $k > \ell \geq 3$, such that S_ℓ has central node c_1 with independent set I_ℓ , S_k has central node c_2 with independent set I_k , $|I_\ell \cap I_k| \geq 2$, and $|I_\ell \setminus I_k| = 1$. A *binary star graph* $B_{\ell k}$ is the union of S_ℓ and S_k . Figure 11 shows some topologies of binary star graphs.

A *binary star subgraph* is an induced subgraph isomorphic to a binary star graph.

Theorem 5 Let $B_{\ell k}$ be a binary star subgraph, and consider the corresponding binary star inequality:

$$\sum_{v \in I_k \cup I_\ell} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq k + 1. \tag{4}$$

Then:

1. The binary star inequality is valid for P_m .
2. The binary star inequality induces a facet of P_m , except when there is a node v such that $N(v) \cap (I_k \cup I_\ell) = I_k \cup I_\ell$.

Proof 1. For $\ell = 3$ and $k = 4$, the validity can be checked by summing up the claw inequality associated with $S'_3 = B_{34}[\{1, 2, 3, c_1\}]$, the star inequality associated with $S'_4 = B_{34}[\{2, 3, 4, 5, c_2\}]$, and the valid inequality $x_1 + x_4 + x_5 + x_{c_1} \leq 4$, yielding the valid inequality:

$$\sum_{v \in I_3 \cup I_4} x_v + x_{c_1} + x_{c_2} \leq 5 + \left\lfloor \frac{1}{2} \right\rfloor = 5.$$

For $\ell = 3$ and $k \geq 5$, the proof is done by induction. Let $B_{\ell k}^v = B_{\ell k} - v$. Note that $B_{3k}^i, B_{3k}^j, B_{3k}^h$, and $S'_3 = B_{3k}[\{i, j, h, c_2\}]$, for distinct $i, j, h \in I_k \setminus I_\ell$, are subgraphs of B_{3k} . Figure 12 shows subgraphs $B_{3k}^i, B_{3k}^j, B_{3k}^h$, and S'_3 for $i = 4, j = k$, and $h = k + 1$.

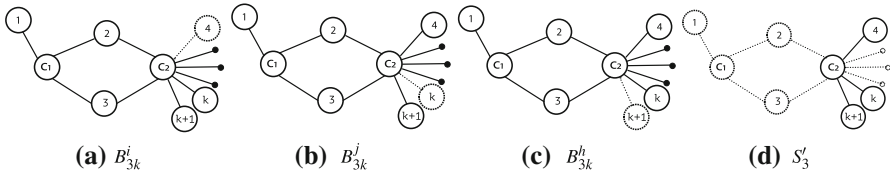


Fig. 12 Subgraphs of B_{3k}

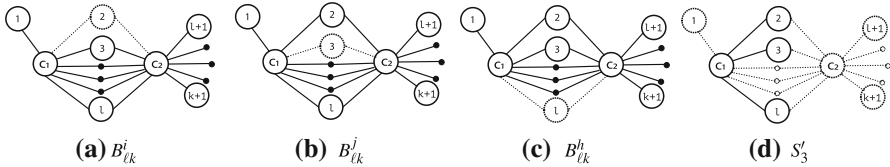


Fig. 13 Subgraphs of $B_{\ell k}$

By induction, the inequalities associated with B_{lk}^i , B_{lk}^j , B_{lk}^h , and S'_3 are valid for P_m , and summing them up leads to

$$3\left(\sum_{v \in I_k \cup I_\ell} x_v\right) + 3(\ell - 2)x_{c_1} + (3k - 3\ell - 2)x_{c_2} \leq 3k + 3.$$

Adding $2x_{c_2} \leq 2$ we get that

$$\sum_{v \in I_k \cup I_\ell} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq (k + 1) + \left\lfloor \frac{2}{3} \right\rfloor = k + 1$$

is valid for P_m .

For $\ell > 3$ and $k \geq \ell + 1$, the proof is also done by induction. Consider the following subgraphs of $B_{\ell k}$: $B_{\ell k}^i$, $B_{\ell k}^j$, $B_{\ell k}^h$, and $S'_3 = B_{\ell k}[\{i, j, h, c_1\}]$, for distinct $i, j, h \in I_\ell$. See Fig. 13, where $i = 2, j = 3$, and $h = \ell$.

By induction, the inequalities associated with $B_{\ell k}^i$, $B_{\ell k}^j$, $B_{\ell k}^h$, and S'_3 are valid for P_m , and summing them up leads to

$$3\left(\sum_{v \in I_\ell \cup I_k} x_v\right) + (3\ell - 6 + 1)x_{c_1} + 3(k - \ell)x_{c_2} \leq 3k + 3.$$

Adding $-x_{c_1} \leq 0$ we get that

$$\sum_{v \in I_\ell \cup I_k} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq k + 1$$

is valid for P_m .

- Let w be a vector such that: $w_{c_1} = \ell - 2, w_{c_2} = k - \ell, w_i = 1$ for $i \in I_k \cup I_\ell$, and $w_v = 0$ for $v \notin V(B_{\ell k})$. Let $u^T x \leq u_0$ be a facet-inducing inequality for the

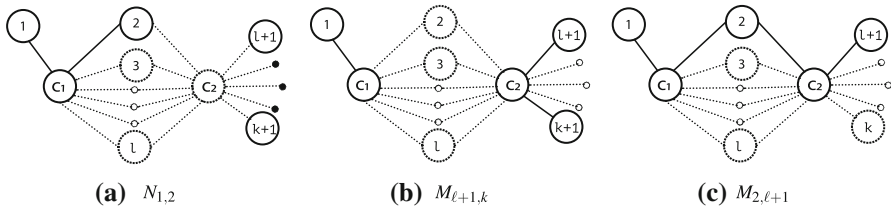


Fig. 14 Subgraphs of $B_{\ell k}$

GDP polytope P_m such that $F_a = \{\chi^H \in P_m \mid w^T \chi^H = k + 1\} \subseteq F_b = \{\chi^H \in P_m \mid u^T \chi^H = u_0\}$. Clearly, $F_a \neq P_m$ and $F_a \neq \emptyset$. Thus, a proof that $u = \alpha w$ for some $\alpha \in \mathbb{R}$ shows that $w^T \chi^H = k + 1$ defines a facet of P_m .

Let $N_{i,j} = B_{\ell k}[(I_k \setminus I_\ell) \cup \{i, j, c_1\}]$ and $M_{i,j} = B_{\ell k}[(I_\ell \setminus I_k) \cup \{i, j, c_1, c_2\}]$ be subgraphs of $B_{\ell k}$ as exemplified in Fig. 14. Observe that $M_{i,j}$ is claw-free when $\{i, j\} \subset I_k \setminus I_\ell$ or $i \in I_\ell \cap I_k$ and $j \in I_k \setminus I_\ell$. Figure 14 shows the graphs $N_{1,2}$, $M_{\ell+1,k}$, and $M_{2,\ell+1}$.

Then:

- a) The incidence vectors $\chi^{N_{i,j}}$ and $\chi^{N_{j,h}}$ with $\{i, j, h\} \subset I_\ell$ are in $F_a \subseteq F_b$. Thus $0 = u_0 - u_0 = u^T \chi^{N_{i,j}} - u^T \chi^{N_{j,h}} = u_i - u_h$. This implies $u_i = u_h = \alpha$ for all choices of $\{i, j, h\} \subset I_\ell$.
- b) The incidence vectors $\chi^{M_{i,j}}$ and $\chi^{M_{j,h}}$ with $\{i, j, h\} \subset I_k \setminus I_\ell$ are in $F_a \subseteq F_b$. Thus $0 = u_0 - u_0 = u^T \chi^{M_{i,j}} - u^T \chi^{M_{j,h}} = u_i - u_h$. This implies $u_i = u_h = \alpha$ for all choices of $\{i, j, h\} \subset I_k \setminus I_\ell$.
- c) The incidence vectors $\chi^{I_k \cup I_\ell}$ and $\chi^{N_{i,j}}$ with $\{i, j\} \subset I_\ell$ are in $F_a \subseteq F_b$. Thus $0 = u_0 - u_0 = u^T \chi^{I_k \cup I_\ell} - u^T \chi^{N_{i,j}} = (\sum_{v \in I_\ell \setminus \{i,j\}} u_v) - u_{c_1}$. This implies $u_{c_1} = \alpha(\ell - 2)$.
- d) The incidence vectors $\chi^{I_k \cup I_\ell}$ and $\chi^{M_{i,j}}$ with $\{i, j\} \subset I_k \setminus I_\ell$ are in $F_a \subseteq F_b$. Thus, $0 = u_0 - u_0 = u^T \chi^{I_k \cup I_\ell} - u^T \chi^{M_{i,j}} = (\sum_{v \in (I_k \cup I_\ell) \setminus \{i,j\}} u_v) - u_{c_1} - u_{c_2}$. This implies $u_{c_2} = \alpha(k - 2) - \alpha(\ell - 2) = \alpha(k - \ell)$.

For a node $v \notin V(B_{\ell k})$ such that $|N(v) \cap (I_k \cup I_\ell)| < 3$, $\chi^{\{v\} \cup I_k \cup I_\ell}$ is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup I_k \cup I_\ell} - u^T \chi^{I_k \cup I_\ell} = u_v$.

For a node $v \notin V(B_{\ell k})$ such that $|N(v) \cap (I_k \cup I_\ell)| > 3$, we have that:

- a) If there is no vertex $i \in N(v) \cap (I_k \setminus I_\ell)$, the incidence vector $\chi^{\{v\} \cup V(N_{i,j})}$, with $i \in N(v) \cap I_\ell$ and $j \in I_\ell$, is in $F_a \subseteq F_b$. Then, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(N_{i,j})} - u^T \chi^{N_{i,j}} = u_v$.
- b) If there are vertices $i \in N(v) \cap (I_k \setminus I_\ell)$ and $j \in (I_\ell \cap I_k) \setminus N(v)$, the incidence vector $\chi^{\{v\} \cup V(M_{i,j})}$ is in $F_a \subseteq F_b$. Therefore, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(M_{i,j})} - u^T \chi^{M_{i,j}} = u_v$.
- c) If there is a vertex $i \in N(v) \cap (I_k \setminus I_\ell)$ and, in addition, $(I_\ell \cap I_k) \setminus N(v) = \emptyset$ and $N(v) \cap (I_k \cup I_\ell) \neq I_k \cup I_\ell$, the incidence vector $\chi^{\{v\} \cup V(M_{i,j})}$, with $i \in N(v) \cap (I_k \setminus I_\ell)$ and $j \in (I_k \setminus I_\ell) \setminus N(v)$, is in $F_a \subseteq F_b$. Thus, the following equality holds: $0 = u_0 - u_0 = u^T \chi^{\{v\} \cup V(M_{i,j})} - u^T \chi^{M_{i,j}} = u_v$.

□

3.4 Formulation for the GDP

A formulation for the GDP is given by combining all the previous facets in the following binary integer programming problem:

$$\begin{aligned}
 \mathcal{F} : \min \quad & \sum_{v \in V(G)} (1 - x_v) \\
 \text{s.t.} \quad & \\
 & \sum_{v \in I_k} x_v + (k - 2)x_c \leq k, && \text{for all star subgraphs } S_k \text{ of } G \\
 & \sum_{v \in I_k} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq k, && \text{for all lantern subgraphs } L_{\ell k} \text{ of } G \\
 & \sum_{v \in I_\ell \cup I_k} x_v + (\ell - 2)x_{c_1} + (k - \ell)x_{c_2} \leq k + 1, && \text{for all binary star subgraphs } B_{\ell k} \text{ of } G \\
 & 0 \leq x_v \leq 1, && \forall v \in V(G) = \{1, \dots, n\} \\
 & x_v \text{ integer}
 \end{aligned}$$

Observe that claw inequalities are star inequalities with $k = 3$.

4 Separation claws, stars, lanterns, and binary stars

\mathcal{F} can have an exponential number of inequalities, therefore a Branch-and-Cut algorithm is suitable for the solution of the GDP. This section explains how to extract support graphs and separate cuts for a Branch-and-Cut algorithm.

4.1 Extracting claw subgraphs

Claw subgraphs are obtained by verifying all triples of mutually nonadjacent vertices that have a common neighbor v in $O(n^4)$ time. Each claw subgraph is added to a collection \mathcal{S}_1 of support graphs.

4.2 Extracting maximal star subgraphs

In order to reduce the number of extracted star subgraphs, we consider only stars formed by a central vertex v and a maximal independent set in $G[N(v)]$ (“maximal stars”). For example, Fig. 15 shows star subgraphs S_3 and S_4 centered at v whose independent sets are maximal in $G[N(v)]$.

Since maximal independent sets of a graph are maximal cliques in its complement, an algorithm for maximal clique extraction *hybrid(.)* proposed by Eppstein and Strash (2011) can be used to extract all maximal stars S_k centered at a vertex v by taking the complement of $G[N(v)]$.

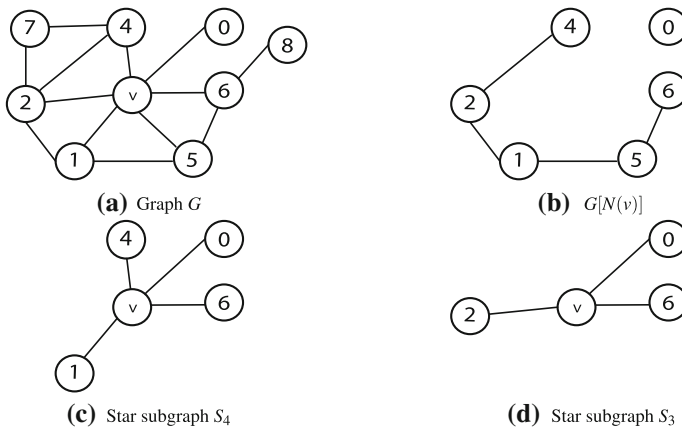


Fig. 15 Extraction of maximal stars centered at v **c** and **d** from **a** by using the neighborhood of v **b**

Algorithm 1 summarizes the extraction of the maximal stars S_k of a graph G . Line 1 initializes the collection \mathcal{S}_2 of support graphs. Line 2 goes through $V(G)$, fixing the center of the star at each iteration. Line 3 computes the complement of $G[N(v)]$. Line 4 extracts all the maximal cliques in $\overline{G[N(v)]}$ using algorithm *hybrid*(\cdot). Lines 5–8 add to \mathcal{S}_2 only maximal stars with independent sets of size at least three.

Algorithm 1: STAR EXTRACTION ALGORITHM

```

1  $\mathcal{S}_2 = \emptyset$ 
2 for  $v \in V(G)$  do
3    $G' = G_n[N(v)]$ 
4    $\mathcal{S}_{temp} = \text{hybrid}(G')$ 
5   for  $I_k \in \mathcal{S}_{temp}$  do
6     if  $k \geq 3$  then
7        $S_k = G[I_k \cup \{v\}]$ 
8        $\mathcal{S}_2 = \mathcal{S}_2 \cup S_k$ 
9 return  $\mathcal{S}$ 

```

4.3 Separation for the basic and star models

With the current collection of support graphs at hand, the Branch-and-Cut algorithm solves the relaxed linear programming problem at each Branch-and-Cut node and then searches for all inequalities violated by the current optimal solution x^* , measures their violation λ , and adds the one hundred most violated inequalities to the current node. The equation $\lambda = \sum_{v \in I_k} x_v^* + (k - 2)x_c^* - k$ is used to measure the violation. If $\lambda > 0,0001$ the inequality is considered to be violated and is added to a pool of inequalities, in order to choose the one hundred best ones (those with the largest values of λ).

4.4 Separation for lantern and binary star models

It is not reasonable to extract all lantern and binary star subgraphs previously to the Branch-and-Cut routines. Thus, a separation heuristic is executed during the Branch-and-Cut algorithm at each node of the tree. It begins by finding the one hundred most violated star inequalities using the same strategy described in Sect. 4.3. Let $\mathcal{S}_{violated}$ be the collection of support star graphs associated with such inequalities. Then, for each pair S_B, S_L of distinct support graphs in $\mathcal{S}_{violated}$, new lantern (binary star) subgraphs are constructed and added to the collection of support lantern (binary star) graphs, and the violation of the associated new inequalities is measured. Finally, the one hundred most violated inequalities are added to the current region See Algorithms 2 and 3 for a brief description of the process used to obtain the new support graphs and associated inequalities. Figures 16 and 17 also illustrate this process. The violation λ for both models is measured as follows:

$$\lambda = \begin{cases} \sum_{v \in I_k} x_v^* + (\ell - 2)x_{c_1}^* + (k - \ell)x_{c_2}^* - k, & \text{for each support lantern graph} \\ \sum_{v \in I_k \cup I_\ell} x_v^* + (\ell - 2)x_{c_1}^* + (k - \ell)x_{c_2}^* - (k + 1), & \text{for each support binary star graph} \end{cases}$$

If $\lambda > 0,0001$ the inequality is considered to be violated.

Algorithm 2: LANTERN SEPARATION ALGORITHM

```

1  $\mathcal{L} = \mathcal{S}_{violated}$ 
2 for distinct  $S_B, S_L \in \mathcal{S}_{violated}$  do
3   Suppose w.l.o.g.  $|V(S_B)| \geq |V(S_L)|$ 
4   Let  $c_B, c_L$  be the centers of  $S_B, S_L$ , respectively
5   Let  $I_B, I_L$  be the independent sets of  $S_B, S_L$ , respectively
6   if  $|I_B \cap I_L| \geq 3$  then
7      $L_1 = G[V(S_B) \cup \{c_L\}]$ 
8      $\mathcal{L} = \mathcal{L} \cup \{L_1\}$ 
9     if  $|I_L| > 3$  then
10       $L_2 = G[V(S_L) \cup \{c_B\}]$ 
11       $\mathcal{L} = \mathcal{L} \cup \{L_2\}$ 
12 Measure  $\lambda$  for all  $L_{\ell k} \in \mathcal{L}$  and add the one hundred best cuts to the current node

```

Algorithm 2 summarizes how to construct the collection \mathcal{L} of support lantern graphs and evaluate the violation of the inequalities associated with such graphs. Line 1 initializes \mathcal{L} with $\mathcal{S}_{violated}$. Lines 2 goes through each possible pair S_B, S_L of stars in $\mathcal{S}_{violated}$. Line 6 checks if the independent sets of S_B and S_L have at least three vertices in common, in order to construct valid lantern graphs. Line 7 creates the support lantern graph L_1 by taking induced subgraph $G[V(S_B) \cup \{c_L\}]$ formed by S_B and the center of S_L , and in line 8 the graph L_1 is stored in the collection \mathcal{L} of support lantern graphs. Lines 9–11 create and store the support lantern graph $L_2 = G[V(S_L) \cup \{c_B\}]$, provided that I_L contains vertices outside $I_B \cap I_L$. Fig. 16 illustrates the construction of L_1 and L_2 .

Algorithm 3: BINARY STAR SEPARATION ALGORITHM

```

1  $\mathcal{B} = S_{violated}$ 
2 for distinct  $S_B, S_L \in S_{violated}$  do
3   Let  $c_B, c_L$  be the centers of  $S_B, S_L$ , respectively
4   Let  $I_B, I_L$  be the independent sets of  $S_B, S_L$ , respectively
5   if  $|I_B \cap I_L| \geq 2$  then
6     for  $v \in V(G) \setminus (I_B \cap I_L)$  do
7       if  $v \in I_L$  and  $|I_B| > |I_B \cap I_L| + 1$  then
8          $B_v = G[V(S_B) \cup \{v\} \cup \{c_L\}]$ 
9          $\mathcal{B} = \mathcal{B} \cup \{B_v\}$ 
10      if  $v \in I_B$  and  $|I_L| > |I_B \cap I_L| + 1$  then
11         $B_v = G[V(S_L) \cup \{v\} \cup \{c_B\}]$ 
12         $\mathcal{B} = \mathcal{B} \cup \{B_v\}$ 
13 Measure  $\lambda$  for all  $B_{\ell k} \in \mathcal{B}$  and add the one hundred best cuts to the current node

```

Algorithm 3 is analogous to Algorithm 2. For each v outside $I_B \cap I_L$, it tries to construct the binary star subgraph $B_v = G[V(S_B) \cup \{v\} \cup \{c_L\}]$ (if $v \in I_L \setminus I_B$) or $B_v = G[V(S_L) \cup \{v\} \cup \{c_B\}]$ (if $v \in I_B \setminus I_L$). The collection \mathcal{B} stores the support binary star graphs.

5 Computational results

5.1 Used models

Four binary integer programming models are proposed by combining the inequalities previously presented. The *claw model* contains only claw inequalities, the *star model* contains star inequalities associated with maximal stars, the *lantern model* adds lantern inequalities to the star model, and the *binary star model* adds binary star inequalities to the star model.

5.2 Random graph instances

Forty random graph instances generated by Bastos et al. (2016) are used to test the four proposed models, with $n \in \{50, 100\}$. All of the benchmark instances are available at

<https://sites.google.com/view/graphdeclaw/instances>.

5.3 Interval graph instances

A random generator is used to create interval graph instances. It receives the number n of intervals, their maximum length, and a time ruler as input parameters. See Algorithm 4 below.

The algorithm maintains a list of intervals \mathcal{I} during its execution. It creates the initial and end times for each interval i and checks if i intersects any previously created interval j ; if an intersection happens, edge (j, i) is added to $E(G)$.

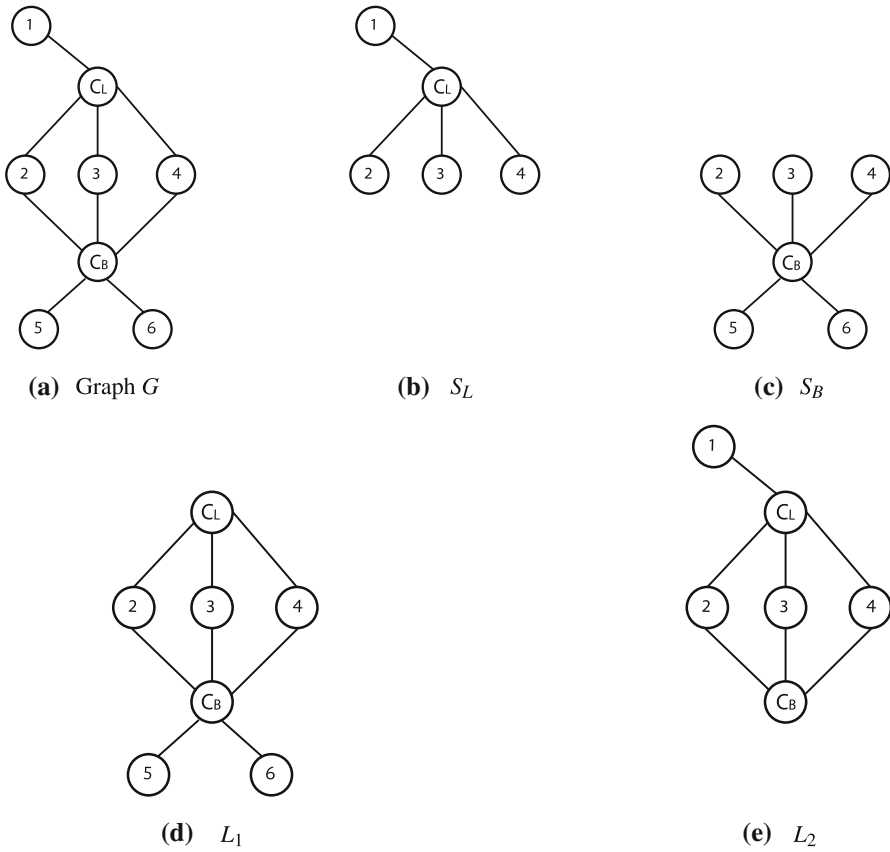


Fig. 16 Figure a shows graph G , figures b and c shows two maximal stars S_L and S_B in G , figures d and e shows the two lantern subgraphs L_1 and L_2 constructed from S_L and S_B

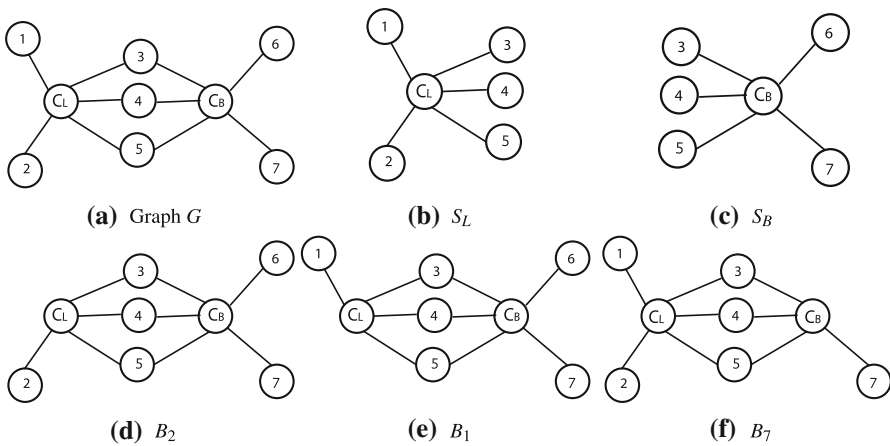


Fig. 17 Figure a shows graph G , figures b and c shows two maximal stars S_L and S_B in G , figures d, e, and f shows three binary star subgraphs constructed from S_L and S_B

Algorithm 4: INTERVAL GRAPH GENERATION ALGORITHM

```

1 Interval_graph_generator ( $n$ ,  $interval\_max\_length$ ,  $time\_ruler$ ):
2    $\mathcal{I} = \emptyset$ ;  $G = \emptyset$ ;  $i = 1$ 
3   for  $i = 1, \dots, n$  do
4      $initial\_time = \text{rand}() \bmod (time\_ruler - interval\_max\_length)$ 
5      $end\_time = initial\_time + (\text{rand}() \bmod interval\_max\_length)$ 
6     if  $initial\_time = end\_time$  then
7        $end\_time = end\_time + 1$ 
8     add interval  $i = (initial\_time, end\_time)$  to  $\mathcal{I}$ 
9     add vertex  $i$  to  $V(G)$ 
10    for  $j \in \mathcal{I}$ ,  $j \neq i$  do
11      if  $(j.initial\_time \leq i.end\_time)$  and  $(i.initial\_time \leq j.end\_time)$  then
12        add edge  $(j, i)$  to  $E(G)$ 
13    return  $G$ 

```

Twenty seven groups of interval graph instances were created by combining the parameters $n \in \{100, 200, 300\}$, $time_ruler = 100$, and $interval_max_length \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$. For each combination (group), nine instances were created. Interval graph instances are named following the pattern x_y_z-d where $x = n$, $y = time_ruler$, $z = interval_max_length$, and d is the identification of the test instance.

5.4 Experimental setup

All algorithms proposed in this work have been developed in C++ with the aid of the mathematical solver CPLEX 11. The computational experiments have been performed on an Intel i7 processor running at 3.4 GHz with 16 GB of RAM, and executing the operating system Linux Ubuntu 14.04.

CPLEX has two internal callbacks to cope with Branch-and-Cut. The first one is the *lazy callback*, which verifies if any integer solution found by the Branch-and-Cut algorithm is feasible. The second one is the *cut callback*, which adds valid cuts to the current node being solved at any time during the execution of the Branch-and-Cut algorithm. For the claw and star models, the lazy and cut callbacks were implemented with the same separation routines described in Sect. 4.3. For the lantern and binary star models, the lazy callback uses the separation routine of the star model, and the cut callback uses proper separation routines described in Sects. 4.4 and 13.

The stop criteria used for the Branch-and-Cut tree are the use of a total memory of 10 Gb and a total time of 6 hours to solve the instances, using only one thread for the processing.

5.5 Comparing the claw, star, lantern, and binary star models for random graph instances

Tables 1 and 2 present results for random graph instances with $n = 50$. All models have been able to solve them optimally. Both tables show the running time, the number of nodes solved, and the number of cuts added by the Branch-and-Cut algorithm for

Table 1 Computational results for random graph instances with $n = 50$

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
105_1_50	0,05	16,00	0,09(0,01)	110	1085	0,02(0,00)	2	337
105_2_50	0,10	19,00	0,18(0,01)	315	747	0,03(0,00)	127	115
105_3_50	0,15	23,00	2,25(0,07)	3415	1650	0,15(0,01)	776	208
105_4_50	0,22	28,00	32,92(0,87)	22593	4351	1,58(0,08)	4073	612
105_5_50	0,24	29,00	116,06(3,40)	61758	5579	3,40(0,17)	6177	913
105_6_50	0,32	32,00	363,81(9,47)	111681	8675	15,19(0,57)	13713	2016
105_7_50	0,32	32,00	278,16(7,77)	86531	9002	21,69(0,93)	20387	2057
105_8_50	0,42	35,00	643,82(17,15)	134288	13036	47,43(1,24)	22035	3694
105_9_50	0,43	35,00	727,60(20,67)	146154	13769	68,22(1,90)	28127	3990
105_10_50	0,47	35,00	666,90(22,94)	136030	13218	57,96(1,55)	21185	4588
106_1_50	0,11	12,00	0,02(0,00)	3	217	0,01(0,00)	4	103
106_2_50	0,16	21,00	0,70(0,02)	972	1246	0,09(0,00)	236	273
106_3_50	0,20	25,00	7,21(0,22)	7254	2810	0,40(0,02)	897	576
106_4_50	0,25	28,00	46,02(1,49)	29146	4913	2,75(0,11)	3853	1142
106_5_50	0,30	31,00	161,31(4,56)	59852	7689	11,08(0,49)	11397	1818
106_6_50	0,32	32,00	209,15(5,31)	65449	8636	9,82(0,35)	8575	2022
106_7_50	0,40	34,00	536,74(15,30)	124642	12034	18,79(0,54)	9289	3382
106_8_50	0,42	35,00	870,03(26,62)	191671	12187	123,19(2,99)	53993	3852
106_9_50	0,44	35,00	712,37(20,09)	147306	13218	67,79(1,83)	28457	4164
106_10_50	0,50	34,00	371,87(13,53)	79052	13039	25,60(0,84)	9194	4797

Table 2 Computational results for random graph instances with $n = 50$

Instances	Lantern model			Binary Star model				
	Density	OPT	T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
105_1_50	0,05	16,00	0,02(0,00)	2	500	0,02(0,00)	3	345
105_2_50	0,10	19,00	0,03(0,00)	127	115	0,03(0,00)	127	115
105_3_50	0,15	23,00	0,14(0,01)	771	211	0,21(0,01)	917	351
105_4_50	0,22	28,00	4,24(0,21)	9214	924	4,88(0,19)	7676	1386
105_5_50	0,24	29,00	5,27(0,23)	7442	1339	7,50(0,25)	7395	2085
105_6_50	0,32	32,00	29,53(0,92)	19662	3186	33,62(0,83)	16529	4205
105_7_50	0,32	32,00	37,33(1,21)	24895	3340	30,25(0,76)	14680	4303
105_8_50	0,42	35,00	80,08(1,41)	21447	6142	81,83(1,32)	17773	7478
105_9_50	0,43	35,00	108,13(2,11)	27877	6580	89,81(1,58)	19757	7262
105_10_50	0,47	35,00	95,15(1,89)	22746	6991	133,74(0,02)	28272	8071
106_1_50	0,11	12,00	0,01(0,00)	4	103	0,01(0,00)	4	103
106_2_50	0,16	21,00	0,08(0,00)	212	307	0,12(0,01)	245	393
106_3_50	0,20	25,00	0,64(0,03)	1304	743	0,77(0,03)	925	1181
106_4_50	0,25	28,00	2,61(0,10)	2734	1495	3,53(0,11)	2365	2336
106_5_50	0,30	31,00	10,90(0,31)	7127	3022	19,09(0,55)	11013	3742
106_6_50	0,32	32,00	15,56(0,46)	10030	2882	17,90(0,38)	7569	4529
106_7_50	0,40	34,00	27,95(0,01)	8615	5731	33,90(0,69)	8351	6715
106_8_50	0,42	35,00	147,42(2,50)	41542	6221	172,75(2,95)	43498	6554
106_9_50	0,44	35,00	86,12(1,61)	23229	6619	101,66(1,80)	23936	7004
106_10_50	0,50	34,00	50,32(0,02)	11254	8268	54,98(1,36)	11063	8271

Table 3 Computational results for random graph instances with $n = 100$

Instances	Claw model				Star model						
	Density	LB	UB	T (s)	#Nodes	#Cuts	LB	UB	T (s)	#Nodes	#Cuts
109_1_100	0,05	45,00	63,00	680,63(211,42)	27062	54678	60,00	60,00	3,54(0,48)	256	6885
109_2_100	0,11	39,51	59,00	596,37(21,99)	100594	11858	55,00	55,00	3296,50(99,59)	3387399	895
109_3_100	0,15	40,75	67,00	554,64(37,07)	46700	25418	56,19	64,00	971,26(49,50)	359124	3470
109_4_100	0,20	40,60	76,00	518,45(51,97)	28998	42083	58,57	72,00	1057,89(55,47)	86873	12149
109_5_100	0,25	41,25	79,00	532,63(70,92)	23391	55790	59,79	75,00	1325,18(56,88)	43298	25334
109_6_100	0,29	41,70	81,00	631,58(87,40)	20251	65533	61,11	78,00	1531,49(59,12)	30837	37710
109_7_100	0,36	42,67	84,00	658,79(122,85)	18571	76663	60,19	82,00	1606,44(69,09)	22971	52368
109_8_100	0,39	43,04	85,00	697,71(131,90)	17215	79505	61,02	83,00	1644,75(70,48)	21432	56833
109_9_100	0,45	44,12	86,00	678,57(159,58)	17296	85094	60,28	85,00	1596,33(71,80)	19387	65034
109_10_100	0,50	44,07	85,00	705,25(158,11)	16198	88850	61,22	85,00	1614,32(70,32)	19117	66042
110_1_100	0,13	43,10	49,00	1647,33(34,45)	253453	8783	48,00	48,00	668,77(15,82)	352227	1742
110_2_100	0,17	41,28	63,00	595,50(40,33)	53454	22135	54,42	59,00	1893,47(62,09)	313549	5195
110_3_100	0,21	41,04	72,00	663,39(49,37)	30194	40564	58,08	69,00	1281,57(57,43)	76838	14558
110_4_100	0,26	41,75	79,00	553,02(66,43)	23599	54720	57,63	77,00	1288,43(56,99)	43005	25475
110_5_100	0,30	42,00	81,00	655,51(85,08)	19422	69145	60,64	79,00	1413,11(55,85)	29670	39072
110_6_100	0,34	42,26	83,00	635,52(111,21)	18460	75576	60,10	82,00	1551,39(65,61)	25240	47224
110_7_100	0,38	43,17	86,00	692,43(124,12)	17623	78625	60,03	83,00	1608,75(69,97)	21903	55566
110_8_100	0,43	43,67	85,00	672,47(142,52)	16945	83828	61,10	83,00	1637,98(72,85)	20265	61687
110_9_100	0,46	43,69	85,00	688,50(160,54)	17271	85066	60,15	85,00	1610,98(72,27)	19061	65935
110_10_100	0,50	44,13	85,00	704,46(155,52)	16141	91062	63,01	84,00	1770,65(70,14)	18597	68311

Table 4 Computational results for random graph instances with $n = 100$

Instances	Lantern model				Binary Star model						
	Density	LB	UB	T (s)	#Nodes	#Cuts	LB	UB	T (s)	#Nodes	#Cuts
109_1_100	0,05	60,00	60,00	9,13(0,66)	206	18805	60,00	60,00	16,05(1,34)	349	18415
109_2_100	0,11	55,00	55,00	2343,30(64,85)	2200393	1100	55,00	55,00	8456,15(119,88)	3632702	2474
109_3_100	0,15	55,43	65,00	900,24(40,13)	271320	4726	55,13	65,00	897,43(31,14)	190931	6640
109_4_100	0,20	58,29	71,00	1179,08(50,73)	72063	17052	57,05	72,00	1088,06(44,26)	58367	20170
109_5_100	0,25	57,10	76,00	1376,73(46,74)	34687	37667	57,70	75,00	1391,88(44,55)	31053	40466
109_6_100	0,29	58,64	78,00	1543,56(52,09)	25496	51650	59,51	77,00	1671,17(49,14)	21923	59654
109_7_100	0,36	60,07	82,00	1762,05(58,26)	18460	75405	60,45	82,00	1801,79(54,12)	16527	83097
109_8_100	0,39	60,97	83,00	1833,58(58,03)	16701	84905	59,98	84,00	1821,07(56,45)	15275	91959
109_9_100	0,45	61,14	85,00	1853,25(60,31)	14924	96929	61,31	84,00	2065,95(59,15)	13577	105852
109_10_100	0,50	61,67	84,00	1940,53(58,82)	14044	106672	59,73	85,00	1794,74(56,96)	14185	100095
110_1_100	0,13	48,00	48,00	717,94(16,36)	369592	1829	48,00	48,00	743,66(11,39)	252114	2716
110_2_100	0,17	54,69	59,00	2377,35(64,99)	327720	6285	53,44	61,00	1221,98(38,00)	173841	7671
110_3_100	0,21	57,91	69,00	1338,68(51,09)	67397	18348	57,37	69,00	1243,53(45,09)	56163	21658
110_4_100	0,26	57,95	77,00	1371,90(47,30)	35000	34661	58,57	77,00	1344,79(45,05)	32580	36849
110_5_100	0,30	59,24	78,00	1635,61(51,79)	24497	54099	59,17	79,00	1622,86(48,13)	21566	59856
110_6_100	0,34	59,17	82,00	1666,62(55,66)	20077	65862	59,47	81,00	1622,35(52,66)	18285	73634
110_7_100	0,38	61,10	83,00	1799,67(57,76)	17132	81057	58,30	83,00	1694,40(57,24)	15973	85594
110_8_100	0,43	59,97	84,00	1800,39(63,49)	15497	94666	59,68	85,00	1712,03(58,01)	15071	94293
110_9_100	0,46	60,96	84,00	1839,95(57,54)	14692	100936	60,93	85,00	1840,75(56,44)	13871	103703
110_10_100	0,50	61,07	84,00	1912,82(57,53)	14094	106656	61,63	84,00	1880,32(54,19)	13070	107612

Table 5 Computational results for interval graph instances not solved to optimality

Instances	Claw model					Star model					
	Density	LB	UB	T (s)	#Nodes	#Cuts	LB	UB	T (s)	#Nodes	#Cuts
200_100_20-1	0.24	66.00	74.00	2034.05(170.92)	36027	40512	71.00	71.00	111.14(10.36)	2620	22471
200_100_20-2	0.23	65.25	73.00	1834.78(155.62)	36531	42987	72.00	72.00	749.30(38.80)	11796	30244
200_100_20-3	0.24	63.02	76.00	1278.98(154.39)	25503	56054	73.00	73.00	880.04(41.02)	10717	35238
200_100_20-4	0.25	64.51	76.00	1686.41(178.19)	26857	53606	72.00	72.00	251.89(22.30)	3778	35330
200_100_20-5	0.22	64.50	82.00	1386.28(149.89)	23664	53248	77.00	77.00	964.83(39.80)	8447	43613
200_100_20-6	0.23	64.51	74.00	1606.42(152.13)	38307	41065	72.00	72.00	2063.31(97.87)	27343	35161
200_100_20-9	0.24	64.39	81.00	1673.53(158.20)	24106	52477	74.00	74.00	1144.33(50.63)	10311	40877
200_100_30-4	0.36	65.00	73.00	3169.96(404.59)	21809	76124	72.00	72.00	526.27(32.61)	3379	51897
300_100_10-1	0.11	87.03	104.00	1238.27(84.14)	38769	28306	100.00	100.00	1634.40(110.16)	32404	20011
300_100_10-2	0.11	81.86	115.00	1338.51(80.68)	33456	33231	94.81	107.00	1743.09(214.21)	38686	30236
300_100_10-3	0.11	87.38	117.00	1322.46(79.72)	31354	33589	99.33	113.00	1849.01(238.41)	40496	28361
300_100_10-4	0.10	86.26	105.00	1393.94(85.41)	38921	28885	101.00	101.00	6699.80(407.60)	136886	22858
300_100_10-5	0.11	85.36	107.00	1092.95(94.91)	36853	29062	97.35	102.00	3603.79(385.95)	72159	24523
300_100_10-6	0.11	84.25	110.00	1115.63(101.55)	35754	30432	93.73	108.00	1685.14(219.17)	36830	31026
300_100_10-7	0.11	85.50	113.00	1458.77(81.19)	31898	33532	100.82	106.00	3240.54(329.69)	61706	25918
300_100_10-8	0.11	82.50	107.00	1307.44(73.40)	38752	28258	94.23	99.00	3198.63(288.53)	60986	24908
300_100_10-9	0.10	85.60	114.00	1350.88(80.06)	34688	30690	97.82	107.00	2176.65(213.80)	43763	28088
300_100_20-1	0.23	87.25	138.00	3322.03(406.15)	10236	117088	104.44	137.00	4543.62(254.13)	8939	134686
300_100_20-2	0.24	89.91	130.00	3573.12(387.06)	10926	111728	106.95	126.00	5276.96(260.27)	11086	113284

Table 5 continued

Instances	Claw model				Star model						
	Density	LB	UB	T (s)	#Nodes	#Cuts	LB	UB	T (s)	#Nodes	#Cuts
300_100_20-3	0.24	88.51	131.00	3397.13(410.17)	10675	115273	106.78	127.00	4769.55(282.55)	10287	122609
300_100_20-4	0.24	88.17	136.00	3075.90(326.58)	9854	122795	109.28	124.00	5871.60(334.99)	11221	116440
300_100_20-5	0.24	88.77	142.00	3538.09(485.36)	9405	127700	108.06	136.00	6142.51(321.06)	7862	156232
300_100_20-6	0.22	84.88	127.00	2892.44(328.80)	12542	97161	103.23	124.00	4687.96(199.84)	10451	120772
300_100_20-7	0.24	87.50	128.00	3300.29(412.23)	10854	107685	115.00	115.00	10634.78(339.91)	15259	103696
300_100_20-8	0.24	89.63	132.00	3467.63(408.72)	9788	118011	111.96	124.00	6524.05(378.29)	11529	114525
300_100_20-9	0.23	86.25	140.00	2954.84(329.00)	10136	117450	108.13	127.00	6145.07(295.07)	9689	129303
300_100_30-1	0.37	90.01	110.00	5212.03(1231.42)	8628	153587	108.00	108.00	2153.47(448.59)	1800	136634
300_100_30-2	0.38	87.67	120.00	5908.49(1162.00)	8076	149072	115.14	117.00	21608.28(1755.68)	12600	201256
300_100_30-3	0.36	93.13	122.00	7066.39(1100.16)	7599	161233	117.00	117.00	1057.29(175.00)	923	101502
300_100_30-4	0.36	85.00	118.00	4820.33(1077.95)	10050	131147	99.69	115.00	6548.15(944.36)	9981	151365
300_100_30-5	0.39	93.00	127.00	7478.74(1626.75)	7603	168243	118.00	118.00	19150.25(3800.82)	9786	210441
300_100_30-6	0.39	97.76	112.00	8207.66(1994.37)	12057	112884	111.00	111.00	2036.75(336.03)	2054	111079
300_100_30-7	0.36	96.01	114.00	6462.65(1580.45)	9978	128686	114.00	114.00	2571.46(473.80)	1933	124882
300_100_30-9	0.36	91.00	110.00	5842.56(981.55)	10054	138267	106.00	106.00	980.75(189.24)	1300	92365
300_100_40-6	0.50	104.51	107.00	21605.69(4181.36)	16700	129677	107.00	107.00	1088.44(654.58)	442	83640
300_100_40-9	0.53	99.50	112.00	13800.68(4484.70)	12462	126926	-	-	-	-	-
300_100_50-1	0.66	70.00	70.00	44.76(38.08)	2	7800	-	-	-	-	-
300_100_50-3	0.66	66.00	66.00	45.63(38.79)	2	7800	-	-	-	-	-
300_100_50-7	0.62	80.00	80.00	233.69(186.36)	167	20156	-	-	-	-	-
300_100_50-9	0.66	70.00	70.00	61.50(52.43)	3	9400	-	-	-	-	-

each model. For this set of instances, the star model, when compared with the other models, achieves better running times for all random graph instances.

Comparing only the star and claw models, we observe that the star model has a fewer number of cuts and expanded nodes for all random graph instances. This is an evidence that star inequalities dominate claw inequalities and thus reach optimal solutions faster. For example, for instance 105_6_50, the star model took nearly eight times fewer nodes and four times fewer cuts. This explains the gain in time to find the optimal solution.

The comparison between the star model and the lantern/binary star models is more involved, because for each model there are instances for which it performs well, in terms of number of cuts or number of expanded nodes. But, in general, the star model is still the one with best results. It seems that star inequalities are simpler to manage, and provide a very strong relaxation. For example, instance 106_8_50 generates fewer nodes for the lantern and binary star models, but a worse execution time than the star model. Still for instance 106_8_50, the star model adds fewer cuts – possibly lantern and binary stars inequalities are making the feasible region more difficult to deal with, forcing the Branch-and-Cut algorithm to take more time to find an optimal solution.

Tables 3 and 4 present results for random graph instances with $n = 100$. Both tables show the separation lower bound, upper bound, running time, number of nodes, and number of cuts for each model. In general, the star model is again the one with best results, although the lantern and binary star models have produced a better lower bound for some instances. Probably, this is due to the time increase these inequalities impose on the Branch-and-Cut tree by reducing the number of nodes.

5.6 Comparing the claw and star models for interval graph instances

Only the claw and star models were used to solve interval graph instances, since chordless cycles with four vertices are forbidden structures for interval graphs, and both lantern and binary star subgraphs contains such cycles.

For instances optimally solved by both models, see Table 6. Instances with density higher than 60% were easier to solve (most of them were solved at the root node).

Table 5 shows the instances not solved to optimality by at least one of the models (Table 6). The star model was able to solve optimally all the instances with $n = 200$ and some with $n = 300$. But it could not solve instances 300_100_40-9, 300_100_50-1, 300_100_50-3, 300_100_50-7, and 300_100_50-9, because of the large amount of maximal independent sets. In such cases, the claw model achieves better results.

6 Conclusions

In this paper we studied the Graph Declawing Problem (GDP). Some facet-defining inequalities of the GDP polytope were presented. Separation algorithms were proposed and used into a Branch-and-Cut procedure for solving the GDP. Computational experiments were carried out and the results show that the algorithms are able to solve many interval graph instances with $n \in \{100, 200, 300\}$ using fewer than six hours

of processing and 10 Gb of memory for the Branch-and-Cut tree. The computational results also show that random graph instances were harder to solve. Random graph instances with $n = 50$ were optimally solved but most random graph instances with $n = 100$ were not solved to optimality.

A Computational results for interval graph instances optimally solved

See [Table 6](#)

Table 6 Computational results for interval graph instances optimally solved

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
100_100_10-1	0.12	22.00	0.06(0.00)	36	628	0.04(0.00)	42	595
100_100_10-2	0.10	21.00	0.04(0.00)	7	465	0.02(0.00)	15	255
100_100_10-3	0.11	22.00	0.04(0.00)	12	445	0.02(0.00)	8	372
100_100_10-4	0.11	17.00	0.06(0.00)	4	620	0.03(0.00)	3	533
100_100_10-5	0.10	19.00	0.05(0.00)	2	466	0.03(0.00)	3	383
100_100_10-6	0.11	23.00	0.07(0.00)	52	538	0.04(0.00)	21	450
100_100_10-7	0.10	21.00	0.05(0.00)	13	511	0.03(0.00)	5	328
100_100_10-8	0.11	21.00	0.05(0.00)	32	567	0.03(0.00)	19	392
100_100_10-9	0.11	22.00	0.05(0.00)	14	537	0.02(0.00)	2	345
100_100_20-1	0.23	30.00	0.47(0.03)	146	3096	0.12(0.00)	4	1438
100_100_20-2	0.24	30.00	0.98(0.04)	430	3201	0.17(0.01)	39	1975
100_100_20-3	0.23	30.00	0.70(0.03)	362	2861	0.31(0.02)	105	2354
100_100_20-4	0.25	32.00	1.70(0.09)	538	4583	0.77(0.05)	311	2825
100_100_20-5	0.21	35.00	11.33(0.52)	4321	6095	1.91(0.12)	876	3871
100_100_20-6	0.23	29.00	0.42(0.02)	161	2644	0.19(0.01)	51	2280
100_100_20-7	0.23	34.00	2.65(0.13)	898	5083	0.45(0.03)	141	2867
100_100_20-8	0.23	27.00	0.16(0.01)	22	1783	0.15(0.01)	18	1799
100_100_20-9	0.22	32.00	1.40(0.05)	484	3654	0.26(0.01)	66	2294
100_100_30-1	0.43	37.00	20.45(2.16)	1997	13645	2.94(0.28)	249	7672

Table 6 continued

Instances	Density	OPT	Claw model		Star model			
			T (s)	#Nodes	T (s)	#Nodes		
100_100_30-2	0.38	29.00	0.47(0.03)	44	3344	0.18(0.01)	2	1606
100_100_30-3	0.36	30.00	0.88(0.08)	161	4261	0.30(0.02)	11	2602
100_100_30-4	0.33	27.00	0.51(0.04)	90	3929	0.19(0.01)	16	1982
100_100_30-5	0.39	34.00	2.92(0.40)	287	8772	0.95(0.11)	49	5797
100_100_30-6	0.41	28.00	0.56(0.07)	57	4729	0.18(0.02)	13	2078
100_100_30-7	0.33	31.00	2.51(0.24)	630	6330	0.40(0.03)	50	3329
100_100_30-8	0.38	33.00	1.75(0.19)	177	6473	0.56(0.05)	17	3347
100_100_30-9	0.39	31.00	3.73(0.36)	772	6762	0.75(0.07)	107	4019
100_100_40-1	0.56	27.00	0.56(0.16)	19	4600	0.30(0.06)	4	2200
100_100_40-2	0.51	30.00	1.13(0.30)	96	4572	0.47(0.07)	2	2753
100_100_40-3	0.50	31.00	1.63(0.32)	172	6856	1.08(0.18)	47	6535
100_100_40-4	0.52	33.00	2.99(0.76)	255	7549	1.02(0.19)	39	5259
100_100_40-5	0.54	28.00	0.71(0.16)	28	4364	0.46(0.06)	5	2773
100_100_40-6	0.47	34.00	5.16(1.02)	473	9017	1.10(0.17)	38	5931
100_100_40-7	0.51	35.00	7.06(1.62)	730	8534	0.66(0.12)	23	3731
100_100_40-8	0.51	32.00	2.98(0.62)	275	7664	1.14(0.13)	27	5180
100_100_40-9	0.50	35.00	6.71(1.54)	513	12294	1.27(0.25)	53	5569
100_100_50-1	0.73	14.00	0.08(0.02)	0	1100	0.09(0.02)	0	900
100_100_50-2	0.60	25.00	0.58(0.18)	37	4356	0.37(0.06)	2	2660

Table 6 continued

Instances	Density	OPT	Claw model		Star model			
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
100_100_50-3	0.69	16.00	0.12(0.05)	0	1500	0.10(0.02)	0	800
100_100_50-4	0.57	22.00	0.25(0.07)	2	2100	0.20(0.07)	0	1700
100_100_50-5	0.73	19.00	0.18(0.09)	0	1600	0.14(0.04)	0	1300
100_100_50-6	0.64	25.00	0.34(0.12)	0	2400	0.30(0.12)	0	1800
100_100_50-7	0.58	27.00	0.59(0.16)	5	3227	0.23(0.08)	0	1800
100_100_50-8	0.68	20.00	0.14(0.06)	0	1900	0.14(0.04)	0	1769
100_100_50-9	0.66	26.00	1.05(0.43)	64	5321	0.55(0.14)	2	3000
100_100_60-1	0.85	10.00	0.06(0.01)	0	800	0.05(0.00)	0	500
100_100_60-2	0.82	12.00	0.13(0.05)	1	1400	0.09(0.02)	0	1000
100_100_60-3	0.83	14.00	0.12(0.03)	4	1200	0.07(0.01)	0	800
100_100_60-4	0.78	13.00	0.11(0.04)	0	1600	0.09(0.02)	0	1000
100_100_60-5	0.86	9.00	0.08(0.02)	2	900	0.05(0.01)	0	700
100_100_60-6	0.83	9.00	0.05(0.01)	0	600	0.05(0.01)	0	600
100_100_60-7	0.79	13.00	0.09(0.03)	0	1200	0.07(0.02)	0	900
100_100_60-8	0.78	11.00	0.07(0.02)	0	1100	0.05(0.01)	0	600
100_100_60-9	0.78	10.00	0.08(0.02)	0	800	0.07(0.01)	0	700
100_100_70-1	0.89	5.00	0.05(0.00)	0	500	0.04(0.00)	0	300
100_100_70-2	0.90	7.00	0.05(0.01)	0	600	0.04(0.00)	0	400
100_100_70-3	0.87	10.00	0.08(0.01)	4	700	0.04(0.00)	0	500

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
100_100_70-4	0.82	8.00	0.06(0.01)	0	800	0.05(0.01)	0	800
100_100_70-5	0.85	7.00	0.06(0.01)	0	600	0.04(0.01)	0	518
100_100_70-6	0.88	5.00	0.04(0.00)	0	300	0.04(0.00)	0	300
100_100_70-7	0.90	6.00	0.05(0.00)	0	600	0.03(0.00)	0	300
100_100_70-8	0.86	7.00	0.05(0.01)	0	800	0.04(0.00)	0	400
100_100_70-9	0.83	6.00	0.06(0.01)	0	500	0.04(0.00)	0	300
100_100_80-1	0.95	1.00	0.04(0.00)	0	100	0.02(0.00)	0	100
100_100_80-2	0.93	3.00	0.04(0.00)	0	200	0.02(0.00)	0	200
100_100_80-3	0.93	4.00	0.04(0.00)	0	200	0.02(0.00)	0	200
100_100_80-4	0.90	4.00	0.04(0.00)	0	400	0.03(0.00)	0	200
100_100_80-5	0.93	4.00	0.04(0.00)	0	300	0.02(0.00)	0	100
100_100_80-6	0.92	5.00	0.05(0.00)	0	400	0.03(0.00)	0	400
100_100_80-7	0.91	6.00	0.05(0.00)	3	400	0.03(0.00)	0	400
100_100_80-8	0.94	3.00	0.04(0.00)	0	200	0.02(0.00)	0	200
100_100_80-9	0.91	5.00	0.04(0.00)	0	300	0.02(0.00)	0	200
100_100_90-1	0.98	0.00	0.04(0.00)	0	0	0.01(0.00)	0	0
100_100_90-2	0.97	1.00	0.04(0.00)	0	100	0.02(0.00)	0	100
100_100_90-3	0.95	2.00	0.04(0.00)	0	200	0.02(0.00)	0	200
100_100_90-4	0.98	1.00	0.04(0.00)	0	100	0.01(0.00)	0	100
100_100_90-5	0.98	0.00	0.04(0.00)	0	0	0.02(0.00)	0	0
100_100_90-6	0.98	0.00	0.04(0.00)	0	0	0.02(0.00)	0	0
100_100_90-7	0.96	0.00	0.04(0.00)	0	0	0.02(0.00)	0	0
100_100_90-8	0.99	0.00	0.04(0.00)	0	0	0.02(0.00)	0	0

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
100_100_90-9	0.95	3.00	0.04(0.00)	0	200	0.02(0.00)	0	200
200_100_10-1	0.10	64.00	1739.87(36.54)	222770	7980	50.27(3.95)	11273	5515
200_100_10-2	0.11	59.00	264.09(5.98)	42434	6584	7.05(0.83)	2566	4468
200_100_10-3	0.11	58.00	14.09(0.57)	4204	5745	0.98(0.11)	170	4155
200_100_10-4	0.10	54.00	11.31(0.53)	4391	5477	2.37(0.20)	754	3717
200_100_10-5	0.11	53.00	3.66(0.24)	1419	4855	0.43(0.05)	94	2762
200_100_10-6	0.11	57.00	110.91(3.61)	31488	6020	6.84(0.70)	2379	4887
200_100_10-7	0.11	66.00	11108.05(198.95)	1123275	9034	277.40(20.01)	56740	5799
200_100_10-8	0.11	55.00	22.71(0.94)	7160	5655	2.10(0.17)	510	4089
200_100_10-9	0.11	61.00	149.70(3.45)	22209	7201	23.05(2.12)	6574	5143
200_100_20-7	0.24	73.00	12124.31(355.81)	68427	54986	111.77(9.87)	1598	25471
200_100_20-8	0.24	72.00	6868.43(249.22)	57879	41997	127.89(10.88)	2077	27212
200_100_30-1	0.36	73.00	3637.97(345.51)	18592	53877	52.03(8.60)	332	28733
200_100_30-2	0.38	69.00	1524.23(140.71)	8295	54410	55.79(9.36)	406	29103
200_100_30-3	0.38	71.00	1033.69(132.92)	7326	41407	42.87(8.88)	420	21181
200_100_30-5	0.38	75.00	10623.76(746.76)	33921	82695	177.12(32.72)	1033	44690
200_100_30-6	0.40	69.00	695.09(93.03)	4574	38305	24.48(5.21)	113	20301
200_100_30-7	0.35	72.00	1743.49(224.45)	9509	55409	89.24(19.48)	536	35222
200_100_30-8	0.39	68.00	316.24(65.14)	2906	34547	8.74(1.96)	24	9449
200_100_30-9	0.37	67.00	992.39(82.99)	8235	43334	72.60(11.97)	643	33810

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
200_100_40-1	0.47	76.00	5448.62(690.29)	13836	79218	208.30(87.08)	859	40355
200_100_40-2	0.51	63.00	145.49(67.80)	1345	21445	28.21(12.26)	10	12911
200_100_40-3	0.53	59.00	25.56(12.75)	206	14870	9.56(3.69)	15	10209
200_100_40-4	0.56	69.00	379.56(148.43)	2334	32692	31.78(11.54)	19	13802
200_100_40-5	0.55	59.00	28.44(16.79)	200	12800	17.05(8.00)	6	10448
200_100_40-6	0.50	72.00	2311.61(363.05)	9032	57260	190.61(57.30)	856	52817
200_100_40-7	0.51	68.00	594.76(169.30)	3396	38716	27.87(11.44)	71	16872
200_100_40-8	0.52	67.00	469.21(164.48)	2731	38712	25.69(11.05)	29	17154
200_100_40-9	0.53	75.00	5305.61(796.45)	16385	71752	222.80(68.72)	784	55488
200_100_50-1	0.68	45.00	10.64(8.13)	7	6500	24.90(14.38)	0	4000
200_100_50-2	0.66	43.00	6.88(4.97)	4	5700	4.84(2.71)	0	3700
200_100_50-3	0.69	39.00	4.80(3.67)	7	4300	8.50(4.91)	0	3800
200_100_50-4	0.61	46.00	10.91(8.85)	2	5800	17.55(11.04)	0	5600
200_100_50-5	0.71	39.00	4.42(3.40)	0	4000	6.76(3.91)	0	3800
200_100_50-6	0.63	51.00	7.96(5.55)	3	5700	10.61(6.50)	0	5000
200_100_50-7	0.61	53.00	22.02(15.04)	40	10636	23.97(15.90)	0	6300
200_100_50-8	0.67	42.00	4.84(3.69)	0	4800	3.84(2.17)	0	4100
200_100_50-9	0.68	46.00	10.38(7.76)	4	6400	13.72(8.58)	0	4500
200_100_60-1	0.79	26.00	1.76(0.99)	0	2200	1.49(0.65)	0	2000

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
200_100_60-2	0.82	28.00	4.32(3.22)	2	3800	5.45(2.92)	0	2800
200_100_60-3	0.77	26.00	2.89(2.04)	0	2900	3.63(1.90)	0	2800
200_100_60-4	0.79	24.00	2.27(1.47)	0	2900	1.26(0.49)	0	1600
200_100_60-5	0.82	23.00	2.26(1.40)	2	2800	1.73(0.75)	0	1900
200_100_60-6	0.84	19.00	1.21(0.51)	0	1600	0.80(0.26)	0	1300
200_100_60-7	0.80	32.00	4.17(2.95)	5	4700	3.93(2.14)	0	3000
200_100_60-8	0.77	20.00	1.48(0.76)	0	2100	1.05(0.40)	0	1500
200_100_60-9	0.79	29.00	2.64(1.79)	0	3100	2.00(0.97)	0	2300
200_100_70-1	0.85	17.00	1.63(0.89)	0	1900	1.16(0.45)	0	1500
200_100_70-2	0.89	12.00	0.89(0.24)	0	1500	0.42(0.10)	0	800
200_100_70-3	0.84	23.00	2.05(1.20)	8	2200	1.76(0.79)	0	1700
200_100_70-4	0.84	15.00	1.36(0.66)	0	2200	0.82(0.28)	0	1100
200_100_70-5	0.85	16.00	1.45(0.72)	0	1800	1.12(0.41)	0	1500
200_100_70-6	0.87	13.00	0.88(0.22)	0	1400	0.50(0.14)	0	1100
200_100_70-7	0.92	11.00	0.82(0.18)	0	1400	0.34(0.08)	0	700
200_100_70-8	0.86	16.00	1.21(0.51)	0	1900	0.76(0.25)	0	1300
200_100_70-9	0.84	14.00	1.17(0.47)	0	1600	0.82(0.30)	0	1500
200_100_80-1	0.90	14.00	0.97(0.28)	4	1600	0.46(0.14)	0	1000
200_100_80-2	0.92	8.00	0.75(0.12)	0	800	0.31(0.06)	0	500

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
200_100_80-3	0.93	7.00	0.64(0.03)	0	400	0.23(0.04)	0	600
200_100_80-4	0.90	8.00	0.69(0.06)	0	700	0.26(0.05)	0	500
200_100_80-5	0.92	10.00	0.76(0.12)	3	800	0.27(0.06)	0	800
200_100_80-6	0.93	8.00	0.71(0.09)	0	700	0.25(0.04)	0	500
200_100_80-7	0.93	8.00	0.67(0.06)	0	600	0.25(0.04)	0	500
200_100_80-8	0.92	11.00	0.78(0.13)	0	1000	0.36(0.09)	0	900
200_100_80-9	0.92	9.00	0.75(0.11)	2	1100	0.26(0.05)	0	500
200_100_90-1	0.96	2.00	0.59(0.00)	0	200	0.16(0.01)	0	200
200_100_90-2	0.96	3.00	0.59(0.00)	0	200	0.15(0.01)	0	200
200_100_90-3	0.96	3.00	0.60(0.00)	0	200	0.15(0.01)	0	200
200_100_90-4	0.97	4.00	0.61(0.00)	0	300	0.14(0.01)	0	300
200_100_90-5	0.99	1.00	0.59(0.00)	0	100	0.12(0.00)	0	100
200_100_90-6	0.98	0.00	0.59(0.00)	0	0	0.12(0.00)	0	0
200_100_90-7	0.97	4.00	0.61(0.01)	0	300	0.16(0.02)	0	300
200_100_90-8	0.98	1.00	0.59(0.00)	0	100	0.13(0.00)	0	100
200_100_90-9	0.97	3.00	0.59(0.00)	0	200	0.14(0.01)	0	200
300_100_30-8	0.37	99.00	2603.03(485.71)	4502	73133	151.19(47.30)	101	41264
300_100_40-1	0.54	92.00	707.34(408.74)	1035	47910	176.34(109.72)	8	18742
300_100_40-2	0.52	89.00	559.04(395.71)	1192	32296	193.20(133.66)	2	16900

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
300_100_40-3	0.52	92.00	1545.94(368.24)	1321	104765	161.33(102.55)	33	23800
300_100_40-4	0.57	98.00	2934.19(1539.56)	5102	49422	154.01(91.30)	11	19300
300_100_40-5	0.55	89.00	322.71(234.56)	464	30100	380.51(256.80)	2	24600
300_100_40-7	0.53	102.00	11961.18(3536.73)	11038	98403	291.06(177.96)	61	31011
300_100_40-8	0.54	104.00	6258.65(2470.50)	7155	67871	1286.73(689.74)	255	106531
300_100_50-2	0.70	63.00	49.56(43.54)	9	7900	86.22(57.25)	0	6500
300_100_50-4	0.61	69.00	66.23(57.11)	6	10200	180.22(124.37)	0	8300
300_100_50-5	0.69	65.00	35.13(30.15)	0	6900	98.94(66.27)	0	7200
300_100_50-6	0.62	81.00	112.54(89.12)	38	16800	274.04(204.17)	0	11900
300_100_50-8	0.66	65.00	37.81(32.84)	0	7200	43.65(27.78)	0	6200
300_100_60-1	0.82	35.00	15.45(11.58)	0	4400	14.16(8.17)	0	3500
300_100_60-2	0.82	43.00	23.77(19.42)	4	4500	44.37(27.80)	0	4700
300_100_60-3	0.79	42.00	31.73(27.18)	2	5200	57.32(36.06)	0	4400
300_100_60-4	0.81	34.00	15.98(12.19)	0	3900	13.69(7.85)	0	3400
300_100_60-5	0.84	28.00	11.00(7.34)	0	3000	7.79(3.54)	0	2400
300_100_60-6	0.82	29.00	9.12(5.57)	0	3300	6.86(3.08)	0	2600
300_100_60-7	0.79	47.00	29.33(24.46)	2	5900	43.03(25.07)	0	3400
300_100_60-8	0.77	37.00	15.27(11.53)	0	3800	10.29(5.27)	0	2800
300_100_60-9	0.78	44.00	22.39(18.27)	0	4800	28.06(18.20)	0	4600

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
300_100_70-1	0.87	32.00	15.68(11.58)	6	4600	19.43(11.74)	0	3700
300_100_70-2	0.88	20.00	7.25(3.73)	4	3200	3.53(1.44)	0	1700
300_100_70-3	0.86	29.00	13.34(9.54)	2	3500	13.41(7.66)	0	3000
300_100_70-4	0.85	26.00	11.89(8.19)	0	4700	5.22(2.40)	0	2100
300_100_70-5	0.87	27.00	8.86(5.29)	8	2900	5.69(2.84)	0	2700
300_100_70-6	0.88	20.00	5.85(2.44)	2	2300	3.23(1.23)	0	1800
300_100_70-7	0.91	14.00	4.44(1.20)	0	2000	1.76(0.47)	0	1000
300_100_70-8	0.86	27.00	13.15(9.43)	0	4200	12.31(6.16)	0	2400
300_100_70-9	0.84	25.00	12.81(9.15)	0	3800	9.29(4.55)	0	2300
300_100_80-1	0.92	18.00	5.32(1.98)	5	2600	2.07(0.70)	0	1300
300_100_80-2	0.93	12.00	4.28(1.05)	2	1500	1.40(0.32)	0	700
300_100_80-3	0.94	10.00	3.31(0.24)	0	800	0.91(0.20)	0	900
300_100_80-4	0.92	12.00	3.64(0.54)	0	1600	0.95(0.23)	0	800

Table 6 continued

Instances	Density	OPT	Claw model			Star model		
			T (s)	#Nodes	#Cuts	T (s)	#Nodes	#Cuts
300_100_80-5	0.92	13.00	3.77(0.58)	0	1000	1.49(0.51)	0	1300
300_100_80-6	0.91	14.00	5.17(1.83)	0	2400	2.57(0.68)	0	1000
300_100_80-7	0.93	13.00	3.86(0.65)	0	1400	1.48(0.41)	0	1200
300_100_80-8	0.92	18.00	5.12(1.77)	2	2300	2.78(1.13)	0	1800
300_100_80-9	0.91	20.00	5.54(2.18)	11	2700	2.40(0.99)	0	1800
300_100_90-1	0.97	7.00	3.08(0.06)	2	600	0.50(0.06)	0	600
300_100_90-2	0.96	6.00	3.21(0.12)	0	800	0.59(0.07)	0	400
300_100_90-3	0.96	3.00	3.07(0.02)	0	300	0.52(0.04)	0	300
300_100_90-4	0.97	8.00	3.22(0.12)	3	700	0.61(0.10)	0	600
300_100_90-5	0.99	1.00	3.00(0.00)	0	100	0.42(0.01)	0	100
300_100_90-6	0.98	1.00	3.02(0.00)	0	100	0.40(0.00)	0	100
300_100_90-7	0.96	6.00	3.18(0.11)	0	600	0.61(0.09)	0	500
300_100_90-8	0.98	3.00	3.05(0.01)	0	300	0.42(0.02)	0	200
300_100_90-9	0.97	6.00	3.15(0.09)	0	700	0.55(0.07)	0	400

References

- Aravind NR, Sandeep RB, Sivadasan N (2017) On polynomial kernelization of h -free edge deletion. *Algorithmica* 79(3):654–666
- Bastos L, Ochi LS, Protti F, Subramanian A, Martins IC, Pinheiro RGS (2016) Efficient algorithms for cluster editing. *J Comb Optim* 31(1):347–371
- van Bevern R, Komusiewicz C, Moser H, Niedermeier R (2010) Measuring indifference: unit interval vertex deletion. In: Thilikos DM (ed) *Graph theoretic concepts in computer science*. Springer, Heidelberg, pp 232–243
- Bogart KP, West DB (1999) A short proof that $\text{proper} = \text{unit}$. *Discret Math* 201(1):21–23
- Bonomo-Braberman F, Nascimento JR, Oliveira FS, Souza US, Szwarcfiter JL (2020) Linear-time algorithms for eliminating claws in graphs. In: *Lecture notes in computer science*, Springer, Newyork, pp 14–26
- Broersma H, Fomin FV, Vant Hof P, Paulusma D (2013) Exact algorithms for finding longest cycles in claw-free graphs. *Algorithmica* 65(1):129–145
- Cygan M, Pilipczuk M, Pilipczuk M, van Leeuwen EJ, Wrochna M (2016) Polynomial kernelization for removing induced claws and diamonds. *Theory Comput Syst* 60(4):615–636
- Eppstein D, Strash D (2011) Listing all maximal cliques in large sparse real-world graphs. Springer, Berlin, pp 364–375
- Faudree R, Flandrin E, Ryjáček Z (1997) Claw-free graphs: A survey. *Discrete Math* 164(1):87–147
- Fishburn PC (1985) Interval graphs and interval orders. *Discret Appl Math* 55(2):135–149
- Fomin FV, Saurabh S, Villanger Y (2013) A polynomial kernel for proper interval vertex deletion. *SIAM J Discret Math* 27(4):1964–1976
- Halldrósson M, Kortsarz G, Shachnai H (2003) Sum coloring interval and k -claw free graphs with application to scheduling dependent jobs. *Algorithmica* 37:187–209
- Hermelin D, Mnich M, van Leeuwen EJ (2014) Parameterized complexity of induced graph matching on claw-free graphs. *Algorithmica* 70(3):513–560
- Hermelin D, Mnich M, Leeuwen EJ, Woeginger G (2019) Domination when the stars are out. *ACM Trans Algorithms* 15(2). <https://doi.org/10.1145/3301445>
- van't Hof P, Villanger Y (2013) Proper interval vertex deletion. *Algorithmica* 65(4):845–867
- Hsu WL, Nemhauser GL (1982) A polynomial algorithm for the minimum weighted clique cover problem on claw-free perfect graphs. *Discret Math* 38(1):65–71
- Ke Y, Cao Y, Ouyang X, Li W, Wang J (2018) Unit interval vertex deletion: fewer vertices are relevant. *J Comput Syst Sci* 95:109–121
- Lewis JM, Yannakakis M (1980) The node-deletion problem for hereditary properties is NP-complete. *J Comput Syst Sci* 20(2):219–230
- Martin B, Paulusma D, van Leeuwen EJ (2020) Disconnected cuts in claw-free graphs. *J Comput Syst Sci* 113:60–75
- Minty GJ (1980) On maximal independent sets of vertices in claw-free graphs. *J Comb Theory Series B* 28(3):284–304
- Roberts FS (1969) Indifference graphs. In: *Proof techniques in graph theory (Proc. Second Ann Arbor Graph Theory Conf., Ann Arbor, Mich., 1968)*, Academic Press, USA, pp 139–146
- Roberts FS (1979) Indifference and seriation. *Ann N Y Acad Sci* 328(1):173–182
- Williams VV, Wang JR, Williams R, Yu H (2015) Finding four-node subgraphs in triangle time. In: *Proceedings of the twenty-sixth annual ACM-SIAM symposium on discrete algorithms, society for industrial and applied mathematics, USA, SODA '15*, pp 1671–1680
- Zhang CQ (1988) Hamilton cycles in claw-free graphs. *J Graph Theory* 12(2):209–216

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.