


On the m -clique free interval subgraphs polytope: polyhedral analysis and applications

Mohammed-Albarra Hassan¹ · Imed Kacem¹ · Sébastien Martin¹  · Izzeldin M. Osman²

Published online: 23 April 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract In this paper we study the m -clique free interval subgraphs. We investigate the facial structure of the polytope defined as the convex hull of the incidence vectors associated with these subgraphs. We also present some facet-defining inequalities to strengthen the associated linear relaxation. As an application, the generalized open-shop problem with disjunctive constraints (GOSDC) is considered. Indeed, by a projection on a set of variables, the m -clique free interval subgraphs represent the solution of an integer linear program solving the GOSDC presented in this paper. Moreover, we propose exact and heuristic separation algorithms, which are exploited into a Branch-and-cut algorithm for solving the GOSDC. Finally, we present and discuss some computational results.

Keywords Interval graph · Polyhedral analysis · Branch-and-cut · Clique

1 Introduction

Motivated by some important parallel-machine scheduling applications, this paper considers the problem of finding m -clique free interval subgraphs. Here, the term m -

✉ Sébastien Martin
sebastien.martin@univ-lorraine.fr

Mohammed-Albarra Hassan
mohamed-albarra.hassan-abdeljabbar@univ-lorraine.fr

Imed Kacem
imed.kacem@univ-lorraine.fr

Izzeldin M. Osman
izzeldin@acm.org

¹ Université de Lorraine, LCOMS EA 7306, Metz 57000, France

² Sudan University of Science and Technology, Khartoum, Sudan

clique free means that the subgraph does not contain any clique of size greater than or equal to $m + 1$ where m is a given positive number. Indeed, this property is essential to ensure the feasibility of a schedule on a set of parallel machines of cardinality equal to m . More precisely, the considered scheduling application is related to an extension of the open-shop family Gonzales and Sahni (1976) where jobs are assigned to a set of machines and have to be sequenced under some disjunctive constraints. The term *disjunctive constraint* means that some couples of jobs cannot run at the same time (i.e., the time intervals when these jobs are performed cannot overlap). Such an extension is called *generalized open-shop with disjunctive constraints (GOSDC)*. Given the structure of this scheduling problem, the study of the m -clique free interval subgraphs polytope seems to be of an important utility.

In a more general way, the considered two families of subgraphs (having the interval property or being m -clique free) have attracted the interest of many researchers. One main reason for this increasing interest is that many real-world applications involve solving problems on graphs, which are either interval graphs themselves or are related to interval subgraphs or to cliques in a natural way. Indeed, numerous applications of interval graphs have appeared in the literature including applications to genetic structures, sequential storage and scheduling [see Golubic 2004; Gacias et al. 2010]. The algorithmic aspects and the combinatorial structures of these subgraphs have been the subject of an intensive research for several decades [see Golubic (2004) for instance]. In some applications, interval representations with special properties are required. As an illustration, the application of the m -clique free interval subgraphs arises in the context of scheduling and cloud computing [see Hassan Abdel-Jabbar et al. 2016].

Now, we give a formal description of our problem and we present some notations. All graphs in this paper are simple and have no self-loops. Let $G = (V, E)$ be a graph. An undirected graph G is called an interval graph if its vertices can be put into a one-to-one correspondence with a set of intervals I of a linearly ordered set (like the real line) such that two vertices are connected by an edge of G if their corresponding intervals have nonempty intersection.

In other words, an interval graph is the graph showing intersecting intervals on a line. Thus, we associate a set of intervals $I = \{I_1, \dots, I_n\}$ on a line with the interval graph $G = (V, E)$, where $V = \{1, \dots, n\}$ and two vertices, u and v , are linked by an edge if and only if $I_u \cap I_v \neq \emptyset$.

Clique is a very common structure in many applications since it is composed of a subset of vertices as well as all the possible relationships among them. Therefore, clique detection is playing an important role in various applications, such as social recommendation Mokotoff and Chretienne (2002) and network routing.

More formally, a clique can be defined as follows. Let $G = (V, E)$ be an undirected graph. A clique in G is a subset $S \subset V$ such that for any two vertices $v_i, v_j \in S$ there exists an edge $(v_i, v_j) \in E$.

Using the same notation, an m -clique in G is a clique $S \subset V$ with $|S| = m + 1$. Finally, we recall that a graph is m -clique free if it does not contain any m -clique.

In parallel machines scheduling some jobs in different machines can share any time units, the jobs can be represented with nodes and edges indicate there is a shared time units between jobs in different machines. For instance, if we consider 2 machines with

the same speed and two jobs of length 3, then if the first job starts on the machine 1 at time $t = 0$ and the second one starts on the machine 2 at time $t = 1$, then there exists an edge between the vertices associated with these two jobs. But, if the second job starts on machine 2 at time $t = 4$, then this edge does not exist. Thus, the solution is mathematically formalized as a graph $G = (V, E)$ where V denotes the set of vertices (associated with jobs) and E denotes the set of relationships between vertices (intersections of jobs on the time line). However, when we assign jobs to m parallel machines, the solution is feasible for two types of graphs (i.e., an interval graph or a m -clique free graph).

2 Polyhedral analysis

Let $\mathcal{I} := \{I \subseteq E \mid G[I] \text{ induces an } m\text{-clique free interval graph}\}$. The vector z^I is called the incidence vector associated with I , i.e., $z^I = (z_e^I)_{e \in E}$, where $z_e^I = 1$ if $e \in I$ and $z_e^I = 0$ otherwise. We define the m -Clique Free Interval Subgraph Polytope (MCFISP) as follows:

$$P_{\mathcal{I}}(G, m) := \text{conv}\{z^I \in \{0, 1\}^{|E|} \mid I \in \mathcal{I}\},$$

First, we give the dimension of $P_{\mathcal{I}}(G, m)$.

Proposition 1 *Polytope $P_{\mathcal{I}}(G, m)$ is full dimensional.*

Proof We will exhibit $|E| + 1$ elements of Polytope $P_{\mathcal{I}}(G, m)$ whose incidence vectors are affinely independent.

Consider the sets $I_0 = \emptyset$, $I_e = \{e\}$ for all $e \in E$. Clearly, these sets induce m -clique free interval subgraphs. Moreover, their incidence vectors are affinely independent. Thus, we have $|E| + 1$ vectors of $P_{\mathcal{I}}(G, m)$ affinely independent and the proof is complete. \square

In the following propositions, we prove that the two trivial inequalities define facets.

Proposition 2 *Inequalities $z_e \geq 0$, $e \in E$, define facets of $P_{\mathcal{I}}(G, m)$.*

Proof see the ‘‘Appendix’’. \square

Proposition 3 *Inequalities $z_e \leq 1$, $e \in E$, define facets of $P_{\mathcal{I}}(G, m)$.*

Proof see the ‘‘Appendix’’. \square

2.1 Forbidden subgraphs inequalities

In this section we discuss graph properties in order to identify valid inequalities. Properties on interval graphs have been studied in Lekkekerker and Boland (1962). The authors give all forbidden subgraphs in interval graphs. Indeed, if a graph does not contain any one of the five subgraphs given in Fig. 1, then it is an interval graph. The two first forbidden subgraphs are called *Bipartite Claw* and *Umbrella*. These

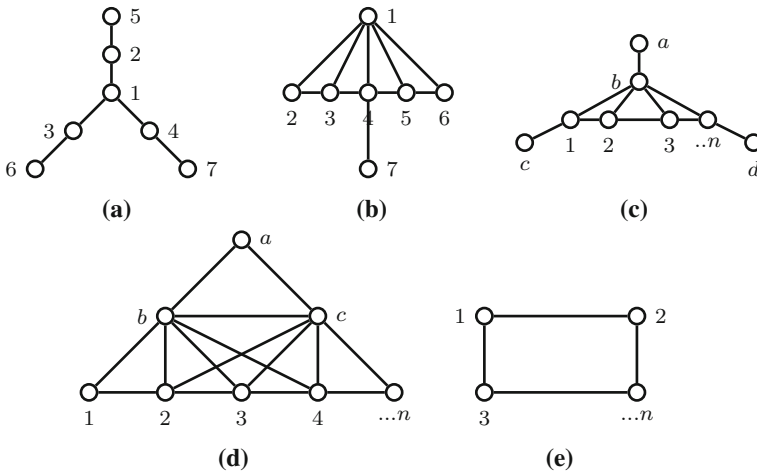


Fig. 1 Forbidden subgraphs characterization. **a** Bipartite Claw. **b** Umbrella. **c** n -net, $n \geq 2$. **d** n -tent, $n \geq 3$. **e** Hole, $n \geq 4$

subgraphs are defined on only seven nodes. The three last forbidden subgraphs are defined on various number of nodes. The n -net subgraphs are composed of $n + 4$ nodes $\{1, \dots, n, a, b, c, d\}$, where the edges are $\{a - b, 1 - c, n - d\} \cup \{1 - b, 2 - b, 3 - b, \dots, n - b\} \cup \{1 - 2, 2 - 3, \dots, (n - 1) - n\}$. The n -tent subgraphs are defined as follows: the nodes $\{a, b, c\}$ are connected by a triangle, the nodes $\{1, \dots, n\}$ are connected in a line form, nodes 1 and b are connected, node n is linked to c and the nodes in $\{b, c, 2, \dots, n - 1\}$ define a clique. Finally, the fifth forbidden configuration is a hole of more than 3 nodes, which is a cycle without chord. These five forbidden subgraphs ensure that the graph is an interval one. In addition, to be m -clique free, we remind that the graph cannot contain any clique of size greater than or equal to $m + 1$.

In the next subsection we describe valid inequalities associated with these forbidden subgraphs and we prove that these inequalities can define facets for $P_I(G, m)$ under some specific conditions.

2.1.1 Bipartite claw

In this subsection, we give inequalities to avoid the forbidden *bipartite claw* subgraph. An example of bipartite claw is given in Fig. 1.

In what follows, we give some notations.

Let us consider the complete graph K_7 with seven nodes. We partition this graph into BC and \overline{BC} , where BC is the set of all edges that form the *bipartite claw* and \overline{BC} is the set of edges in the associated complementary graph of BC . Moreover, \overline{BC} is partitioned as follows:

- Subset \overline{BC}_h^4 contains all the edges such that each of them enables to form a hole of size 4 in a bipartite claw.
- Subset \overline{BC}_Δ contains three edges such that when we add one of them to BC , then we obtain a central triangle.

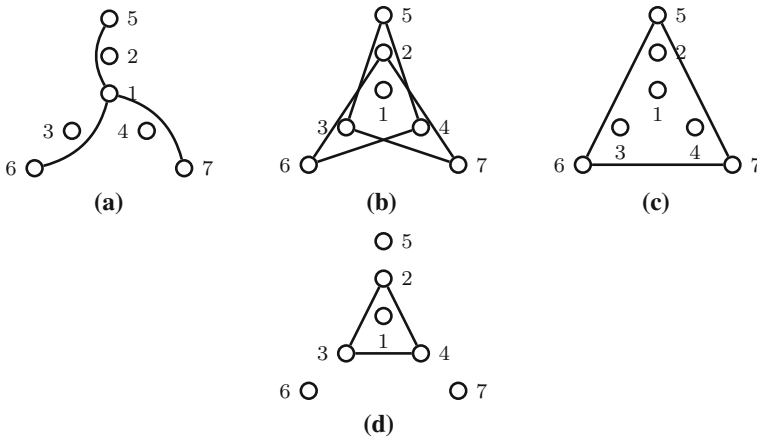


Fig. 2 Subsets of the complementary *Bipartite Claw*. **a** Subset \overline{BC}_i . **b** Subset \overline{BC}_h^4 . **c** Subset \overline{BC}_h^5 . **d** Subset BC_Δ

- Subset \overline{BC}_i contains the edges that are able to form a triangle with the vertex 1.
- Subset \overline{BC}_h^5 is composed of all edges such that each of them enables to form a hole of size 5 in the *bipartite claw*.

Figure 2 shows these subsets.

As a consequence, the previous definitions lead explicitly to the following subsets:

- $BC = \{(1, 2), (1, 3), (1, 4), (2, 5), (4, 7), (3, 6)\}$.
- $\overline{BC} = \{(7, 1), (7, 2), (7, 3), (7, 5), (7, 6), (6, 1), (6, 2), (6, 4), (6, 5), (5, 1), (5, 3), (5, 4), (4, 2), (4, 3), (3, 2)\}$.
- $\overline{BC}_h^4 = \{(3, 5), (2, 6), (5, 4), (2, 7), (3, 7), (4, 6)\}$.
- $\overline{BC}_\Delta = \{(2, 3), (2, 4), (3, 4)\}$,
- $\overline{BC}_i = \{(1, 5), (1, 6), (1, 7)\}$.
- $\overline{BC}_h^5 = \{(5, 6), (5, 7), (6, 7)\}$.

We consider two cases, when $m = 2$, and when $m \geq 3$.

If $m = 2$, then the following inequality is valid:

$$\sum_{e \in BC} z_e \leq 5. \tag{1}$$

It is worthnoting that keeping all the edges of BC leads to the impossibility of respecting the validity of the constraint for this special case when $m = 2$. Indeed, when we add an edge from $\overline{BC}_\Delta \cup \overline{BC}_i$ in Fig. 2d to BC , the resulting subgraph will contain a clique of size 3 and hence would not be an m -clique. Moreover, if we add an edge $e \in \overline{BC}_h^4 \cup \overline{BC}_h^5$ to BC , then we obtain a *hole*. If we add another edge to break this hole, then we obtain a clique of size 3.

Proposition 4 *Inequality (1) defines a facet of $P_{\mathcal{I}}(G, m)$, when $m = 2$.*

Proof see the “Appendix”. □

Now, if $m \geq 3$, then the following inequality is valid.

$$\sum_{e \in BC} z_e - \sum_{e \in \overline{BC}_\Delta} z_e - \sum_{e \in \overline{BC}_i} z_e \leq 5 \tag{2}$$

Indeed, if we add only one edge (respectively two edges) of \overline{BC}_Δ to the bipartite claw, then the resulting subgraph contains 2 – net (respectively 3 – net), which is already excluded by definition (as we mentioned in 2.1). We can observe that if we add one edge of \overline{BC}_Δ , for instance (2, 3), and the edge (2, 6) or (3, 5) from \overline{BC}_h^4 to BC , then we obtain an m -clique free interval subgraph. Furthermore, if we add also the edge (5, 6), then the result remains an m -clique free interval subgraph. It is clear that when we add one, two or three edges of \overline{BC}_i to BC , then the result is an m -clique free interval subgraph. The result is the same when we add (2, 4) and (4, 5) to BC .

Proposition 5 *Inequality (2) defines a facet of $P_{\mathcal{I}}(G, m)$, when $m \geq 3$.*

Proof Let us denote by $az \leq \alpha$ the inequality (2). Let $bz \leq \beta$ be an equality that defines a facet of $P_{\mathcal{I}}(G, m)$, such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. Since $P_{\mathcal{I}}(G, m)$ is full dimensional, we need to prove that there exists ρ such that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $e_1, e_2 \in BC$ be two edges and let us define $I_1 = BC \setminus \{e_1\}$ and $I_2 = BC \setminus \{e_2\}$. It is clear that the incidence vectors of I_1 and I_2 satisfy inequality (2) with equality. Since $az^{I_1} = az^{I_2}, bz^{I_1} = bz^{I_2}$, implying that $b(e_1) = b(e_2)$. We set $b(e_1) = \rho$. As e_1 and e_2 are arbitrary in BC , then $b(e) = b(e') = \rho \forall e, e' \in BC$.

Let $e_3 \in \overline{BC}_i$. We can remark that the incidence vectors of $I_3 = BC \cup \{e_3\}$ and I_1 satisfy inequality (2) with equality. As $az^{I_3} = az^{I_1}$, then $bz^{I_3} = bz^{I_1}$, implying that $b(e_3) = -b(e_1)$. As e_3 is arbitrary in \overline{BC}_i and $e_1 \in BC$, then $b(e) = -b(e'')$ for every $e'' \in \overline{BC}_i$, which leads to $b(e'') = -\rho$.

The incidence vectors of the following three sets $I_4 = BC \cup \{(2, 4), (5, 4)\}, I_5 = BC \cup \{(2, 4), (5, 4), (2, 7)\}$ and $I_6 = BC \cup \{(2, 4), (5, 4), (5, 7)\}$ satisfy inequality (2) with equality. Since $az^{I_4} = az^{I_5} = az^{I_6}$, we have $bz^{I_4} = bz^{I_5} = bz^{I_6}$, which implies that $b((2, 7)) = 0$ and $b((5, 7)) = 0$. By symmetry, we have $b((5, 4)) = b((2, 7)) = b((6, 4)) = b((3, 7)) = b((3, 5)) = b((2, 6)) = b((5, 7)) = b((5, 6)) = b((6, 7)) = 0$.

Similarly, the incidence vectors of the two sets $I_7 = BC \setminus \{(1, 4)\}$ and I_4 verify inequality (2) with equality. Since $az^{I_7} = az^{I_4}, bz^{I_7} = bz^{I_4}$, implying that $b((1, 4)) = -b((2, 4))$. By symmetry, we have $b((2, 4)) = b((3, 4)) = b((2, 3)) = -\rho$.

Let $e_8 \in E \setminus (BC \cup \overline{BC})$. The incidence vectors of sets $I_8 = (BC \setminus \{(1, 2)\}) \cup \{e_8\}$ and $I_9 = BC \setminus \{(1, 2)\}$ satisfy inequality (2) with equality. Since $az^{I_8} = az^{I_9}, bz^{I_8} = bz^{I_9}$, which implies that $b(e_8) = 0$. By symmetry, we have $b(e) = 0 \forall e \in E \setminus (BC \cup \overline{BC})$.

To summarize, there exists ρ such that $b = \rho a$ for some $\rho \in \mathbb{R}$. □

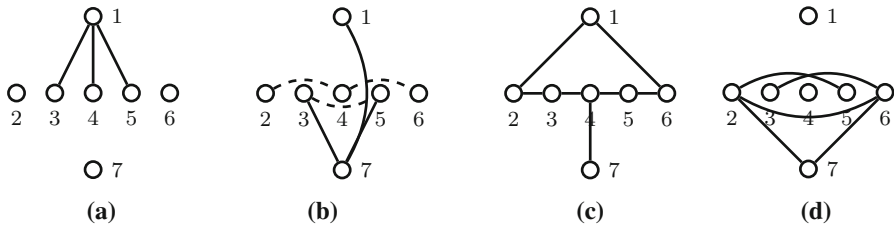


Fig. 3 Subsets of umbrella and its complementary. **a** Set E_u^i . **b** Set $E_u^t \cup E_u^c$. **c** Set E_u^a . **d** Set E_u^h

2.1.2 Umbrella inequalities

For the umbrella subgraph as shown in Fig. 3b, let $G_u = (U_u, E_u)$ be a graph that constitutes the umbrella and let \bar{E}_u be a set of the complementary edges for G_u . In what follows, we will present a family of valid inequalities that avoid the umbrella subgraphs. To analyze this forbidden subgraph we need the following notations:

Let $E_u^i \subset E_u$ be the set of the three edges $\{(1,3), (1,4), (1,5)\}$ in the umbrella subgraph. Let $E_u^t \subset \bar{E}_u$ be the set of the edges such that when we add one of these edges to the umbrella we create a new triangle. Subset E_u^c is the dashed edges in Fig. 5a. Finally, $E_u^a \subset E_u$ is the set of the around edges and $E_u^h \subset \bar{E}_u$ is the set of edges such that if we add one of them to the umbrella, then they will form a hole of size 4 or of size 5. All these subsets are illustrated in Fig. 4.

- $E_u^i = \{ (1, 3), (1, 4), (1, 5) \}$.
- $E_u^t = \{ (1, 7), (3, 7), (5, 7) \}$.
- $E_u^c = \{ (2, 4), (3, 5), (4, 6) \}$.
- $E_u^a = \{ (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (1, 6), (4, 7) \}$.
- $E_u^h = \{ (2, 7), (6, 7), (2, 5), (2, 6), (3, 6) \}$.

It can be observed that the graph induced by $H^u = \{E_u^i \cup E_u^t \cup E_u^c \cup E_u^a \cup E_u^h\}$ is a complete graph.

When $m = 2$, the triangle becomes a forbidden subgraph and the umbrella inequalities are dominated by the inequalities (9) presented in Sect. 2.1.6.

For this forbidden subgraph we focus on the instances for which $m \geq 3$.

When $m = 3$, in order to keep all edges of E_u it is necessary to add at least one edge of E_u^i . Moreover, when we add an edge from E_u^c in this case the subgraph contains a clique of size 4. If we add an edge from E_u^h , then the induced subgraph will contain a hole.

Thus, the valid inequalities when $m = 3$ will be:

$$\sum_{e \in E_u^c \setminus \{(4,7)\}} z_e + z_{(2,6)} + z_{(2,5)} + z_{(3,6)} \leq 5. \tag{3}$$

Proposition 6 Inequality (3) defines a facet of $P_{\mathcal{I}}(G, m)$ if $m = 3$.

Proof see the ‘‘Appendix’’.

□

When $m \geq 4$, to find a feasible solution we can add also the edges from E_u^c . Then, the valid inequalities when $m \geq 4$ will be:

$$\sum_{e \in E_u^a} z_e - \sum_{e \in E_u^t \cup E_u^c} z_e \leq 6. \tag{4}$$

Proposition 7 Inequality (4) defines a facet of $P_{\mathcal{I}}(G, m)$ if $m \geq 4$.

Proof see the ‘‘Appendix’’. □

2.1.3 *n*-net inequalities

The *n* – net forbidden subgraph is shown in Fig. 1c. We will give some notations to help in analyzing the *n* – net forbidden subgraph.

Let $G_{net} = (U_{net}, E_{net})$ be the graph that forms a *net* of size *n* (i.e., *n* – net) and \bar{E}_{net} be the set of complementary edges of G_{net} . To avoid to have a subgraph that represents an *n* – net, where $n \geq 2$ we need either to eliminate an edge from the *n* – net without inducing a hole (this edge does not belong to E_{net}^h), or to add an edge that does not create a hole (this edge does not belong to $E_{net}^{\bar{h}}$).

To analyze this forbidden subgraph we will use the following notations. From Fig. 1c let us consider:

- $E_{net}^{\bar{h}} = \{(a, c), (a, d)\} \cup \{(c, 3), (c, 4), \dots, (c, n)\} \cup \{(d, 1), (d, 2), \dots, (d, n - 2)\} \cup \{(c, d)\}$.
- $E_{net}^h = \{(b, 2), \dots, (b, n - 1)\}$.

We propose valid inequalities that delete the *n* – net forbidden subgraphs.

$$\sum_{e \in E_{net} \setminus E_{net}^h} z_e - \sum_{e \in \bar{E}_{net} \setminus E_{net}^{\bar{h}}} z_e \leq |E_{net} \setminus E_{net}^h| - 1. \tag{5}$$

Proposition 8 Inequality (5) defines a facet of $P_{\mathcal{I}}(G, m)$.

Proof see the ‘‘Appendix’’. □

2.1.4 *n*-tent inequalities

Figure 1d shows the *n* – tent forbidden subgraph. A graph is non-interval if it contains an *n* – tent forbidden subgraph G_{tent} . The subgraph G_{tent} can be transformed into a valid one by adding or removing one or more edges if such modifications do not create a hole.

Let the graph $G_{tent} = (U_{tent}, E_{tent})$ be a graph that constitutes an *n* – tent ($n \geq 3$) and \bar{E}_{tent} be the set of complementary edges.

From Fig. 1c

- $E_{tent}^h = \{(b, c), (c, 4), (b, 2)\}$,
- $E_{tent}^{\bar{h}} = \{(1, 4), (2, 5), \dots, (n, n + 3)\}$.

We can notice that, all n -tent where $n \geq 5$ contain a clique of size 5 then the clique inequality cuts this n -tent subgraph if $m \leq 4$. It is the same idea for all n -tent. Indeed, when $n = 4$ (respectively $n = 3$), the clique inequality cut these n -tent sub graph if $m = 3$ (respectively $m = 2$).

The following inequality allows us to avoid the n -tent forbidden subgraphs.

$$\sum_{e \in E_{tent} \setminus E_{tent}^h} z_e - \sum_{e \in \bar{E}_{tent} \setminus \bar{E}_{tent}^h} z_e \leq |E_{tent} \setminus E_{tent}^h| - 1. \tag{6}$$

Proposition 9 *Inequality (6) defines a facet of $P_{\mathcal{I}}(G, m)$, when $m \geq 5$ or $(n = 4$ and $m = 3)$ or $(n = 3$ and $m = 2)$.*

Proof see the ‘‘Appendix’’. □

2.1.5 Hole inequalities

Here, it is convenient to define a hole as an induced subgraph of G isomorphic to C_k for some $k \geq 4$, Schrijver (2003). The hole C is a forbidden subgraph as depicted in Fig. 1e. Let C denote the set of edges that form the hole, i.e., $C = \{(u_1, u_2), (u_2, u_3), \dots, (u_{|C|-1}, u_{|C|}), (u_{|C|}, u_1)\}$. If i is multiple of $|C|$, then $u_i = u_{|C|}$. Otherwise, if $i > |C|$, then $u_i = u_{i'}$, where i' is the rest of the euclidean division of i by $|C|$. Let \bar{C} denote the set of all chords of hole C .

Suppose we have a hole of size 4. Obviously, this hole is non-interval graph. Nevertheless, the subgraph induced by a hole can be transformed to a feasible one by adding at least one chord.

Proposition 10 *For a hole C , the minimum number of necessary chords that should be added to the hole to be an interval graph is $|C| - 3$, when $|C| \geq 4$.*

Proof We prove this proposition by induction. When $|C| = 4$, then we need one chord $k_c = 1$, and then the proposition is true.

Let us assume that the property is true until $|C| = l$ and $k_c = l - 3$. We need to prove it for a cycle C' where $|C'| = |C| + 1$ and that $k_{c'} = |C'| - 3$. Let $\tilde{C} = \{u_1, u_2, u_3, \dots, u_{|C|}\}$ be the hole graph given by C where we add k_c edges in order to make it hole free and let (u_i, u_{i+1}) be an edge, where $i \in \{1, 2, \dots, |C| - 1\}$. Now, we will construct \tilde{C}' from \tilde{C} by adding the vertex $u_{|C|+1}$ in \tilde{C} , by replacing the edge (u_i, u_{i+1}) by $(u_i, u_{|C|+1})$ and $(u_{|C|+1}, u_{i+1})$. Then, we find a unique hole of a size 4 in the new cycle. Therefore, $|\tilde{C}'| = l + 1$. It is necessary to add one edge to break this hole. Thus, $k_{c'} = l' - 3$ is true. As C and k_c are arbitrary, then the property is true for every hole. □

Now, we will present valid inequalities for the forbidden hole subgraph.

If $m = 2$, then the inequality (7) is valid.

$$\sum_{e \in C} z_e + \sum_{e \in \bar{C}} z_e \leq |C| - 1. \tag{7}$$

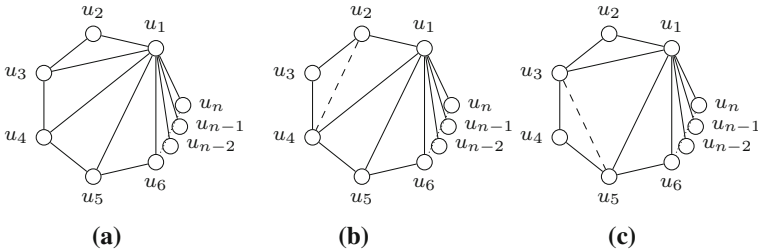


Fig. 4 Hole free subgraphs

Indeed, if we add one chord to hole C , then we will obtain a triangle or another hole and it is not feasible for $m = 2$. It is worth to note that this inequality for $m = 2$ is equivalent to the clique inequalities described in the next subsection.

If $m \geq 3$, then the following inequality is valid.

$$\sum_{e \in C} (|C| - 3)z_e - \sum_{e \in \bar{C}} z_e \leq (|C| - 1)(|C| - 3). \tag{8}$$

Proposition 11 *Let C be a hole of size greater than 3, then inequality (8) associated with cycle C defines a facet of $P_{\mathcal{I}}(G, m)$ if $m \geq 3$.*

Proof Let us denote by $az \leq \alpha$ the inequality (8), associated with hole C . Let $bz \leq \beta$ be a facet defining inequality of $P_{\mathcal{I}}(G, m)$ such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let e_1 and e_2 be two distinct edges of hole C . The solutions $I_1 = C \setminus \{e_1\}$ and $I_2 = C \setminus \{e_2\}$ are m -clique free interval subgraphs. We deduce that I_1, I_2 are solutions. Moreover, we have $az^{I_1} = az^{I_2}$. Hence, $bz^{I_1} = bz^{I_2}$. This implies that $b(e_1) = b(e_2)$. Set $b(e) = (|C| - 3)\rho$. As e_1, e_2 are arbitrary then $b(e) = b(e') \forall e, e' \in C$. We deduce that $b(e) = (|C| - 3)\rho$ for all $e \in C$.

Let $u_i \in V(C)$. The solution $I_3 = C \cup (\delta(u_i) \cap \bar{C})$, illustrated in Fig. 4a, and the solution $I_4 = (I_3 \setminus \{(u_i, u_{i+2})\}) \cup (u_{i+1}, u_{i+3})$, illustrated in Fig. 4b, are feasible and verify the inequality (8) with equality. Moreover, we have $az^{I_3} = az^{I_4}$ and then $bz^{I_3} = bz^{I_4}$. This implies that $b(u_i, u_{i+2}) = b(u_{i+1}, u_{i+3})$. Thus, by symmetry $b((u_i, u_{i+2})) = b((u_{i+1}, u_{i+3}))$ are arbitrary for all $\forall i \in \{1, 2, \dots, |C|\}$.

The solution $I_5 = (I_3 \setminus (u_i, u_{i+j})) \cup (u_{i+j-1}, u_{i+j+1})$, illustrated in Fig. 4c, is feasible and verifies the inequality (8) with equality. Therefore, $az^{I_5} = az^{I_3}$. Hence $bz^{I_5} = bz^{I_3}$. This implies that $b(u_i, u_{i+j}) = b(u_{i+j-1}, u_{i+j+1})$. As these edges are arbitrary then $b(e) = b(e')$ for all $e' \in \bar{C}$.

Now, we will consider the two solutions I_1 and I_3 . These solutions are feasible and verify the inequalities (8) with equality. Hence, $az^{I_1} = az^{I_3}$ and therefore $bz^{I_1} = bz^{I_3}$. This implies that $b(e_1) = -\sum_{e' \in \delta(u_i) \cap \bar{C}} b(e')$. As e_1 and u_i are arbitrary for all $e \in C$ and $u_i \in V(C)$, then $b(e_1) = -(|C| - 3)b(e)$. We deduce that $b(e) = -\rho \forall e \in \bar{C}$.

Let $e_3 \in E \setminus (C \cup \bar{C})$. The solutions $I_7 = I_3 \cup \{e_3\}$ and I_3 are feasible and verify the inequalities (8) with equality. Hence, $az^{I_7} = az^{I_3}$ and then $bz^{I_7} = bz^{I_3}$. This implies that $b(e_3) = 0$. As $b(e_3)$ is arbitrary, then $b(e) = 0$ for all $e \in E \setminus (C \cup \bar{C})$. \square

2.1.6 Clique inequalities

In this section we study the structure of a clique-free subgraph and we propose valid inequalities and facets of $P_{\mathcal{I}}(G, m)$.

Proposition 12 *Let K be a clique and let $V(K)$ be its set of vertices. If $m = 2$, then the following inequality*

$$\sum_{e \in E(K)} z_e \leq |V(K)| - 1 \tag{9}$$

is valid and defines a facet of $P_{\mathcal{I}}(G, m)$.

Proof see the ‘‘Appendix’’. □

Remark that, the inequalities (9) are equivalent to the hole inequalities and dominate the umbrella inequalities when $m = 2$.

Proposition 13 *Let K be a clique of size $m + 1$. The inequality*

$$\sum_{e \in E(K)} z_e \leq |E(K)| - 1 \tag{10}$$

defines a facet of $P_{\mathcal{I}}(G, m)$.

Proof see the ‘‘Appendix’’. □

Let $f(K, m)$ be a function giving the minimum number of edges necessary to be removed from $E(K)$ such that the resulting graph $G'(K)$ is m -clique free. Let $\alpha = \lceil \frac{|V(K)|}{m} \rceil$, $n_{\alpha-1} = m\alpha - |V(K)|$ and $n_\alpha = \frac{|V(K)| - (n_{\alpha-1})(\alpha-1)}{\alpha}$.

Proposition 14 $f(K, m) = n_{\alpha-1} \frac{(\alpha - 1)(\alpha - 2)}{2} + n_\alpha \frac{(\alpha)(\alpha - 1)}{2}$.

Proof Let E' be a set of $f(K, m)$ edges such that $E(K) \setminus E'$ is m -clique free. Note that in the complementary graph $\overline{G}(K)$ we have a stable set of size $|K|$. If we add E' , then there does not exist a stable of size $m + 1$ (m -stable free). Let us consider a minimum set of edges $E'' \in E(K)$ such that $|E''| < |E'|$ and $G(K) \setminus E''$ is m -clique free and thus $\overline{G}(K) \cup E''$ is m -stable free. Clearly E'' is a set of m disjoint cliques $\mathcal{K} = \{K_1, \dots, K_m\}$, otherwise E'' is not minimal or contains a stable set of size $m + 1$. To have the minimum set of edges, it is important to balance the size of cliques. Indeed, if we have two cliques K_i and K_j where $|K_i| \geq |K_j| + 2$, then we can easily prove the following result $|E(K_i)| + |E(K_j)| > |E(K_i \setminus \{u\})| + |E(K_j \cup \{u\})|$, where $u \in K_i$. Thus, it can be deduced that the maximum difference between the cardinalities of two cliques of \mathcal{K} is less or equal to 1. □

Proposition 15 *Let K be a clique. The following inequality*

$$\sum_{e \in E(K)} z_e \leq |E(K)| - f(K, m) \tag{11}$$

is valid for $P_{\mathcal{I}}(G, m)$.

Proof First note that inequality (10) is a special case of inequality (11). However, for the sake of clarity, we prefer to do it step by step. By definition $f(K, m)$ is the minimum number of edges necessary to be removed from $E(K)$ such that the resulting graph $G'(K)$ is m -clique free. Thus, the inequality (11) is valid. \square

In the next subsection, we will improve this family of inequalities.

2.1.7 Clique–Hole inequalities

In order to get an m -clique free subgraph, we need to remove from $G(K)$ a set of m cliques of similar size α or $\alpha - 1$ (as we mentioned in the previous proof). Other holes will remain and they must be avoided. We observe that if we remove m disjoint cliques from $G(K)$, then we obtain a complete bipartite subgraph between all pairs of two cliques. Let $H_{ij} = (K_i, K_j, E_{ij})$ be a complete bipartite graph. Note that H contains a hole of a size greater than or equal to 4 if $|K_i| \geq 2$ and $|K_j| \geq 2$. To remove every hole in H_{ij} , the minimum number of edges E' necessary to be eliminated is equal to $\max(|K_i|, |K_j|) - 1$, otherwise we can always take 2 nodes in K_i or K_j such that these two nodes are not covered by E' . Note that E' , with size $\max(|K_i|, |K_j|) - 1$, covers the maximum nodes of K_i and K_j . We can reinforce the inequality (11) by the following inequality. Let $\alpha = \sum_{i \in m} (\max\{|K_i| - 1, 0\}) - \max_{i \in m} |K_i| - 1$.

Proposition 16 *Let K be a clique. Then, the inequality*

$$\sum_{e \in E(K)} z_e \leq |E(K)| - (f(K, m) + \alpha) \tag{12}$$

is valid for $P_{\mathcal{I}}(G, m)$.

Proof We can notice that if we unbalance two cliques K_i and K_j to reduce the value $\max(|K_i|, |K_j|) - 1$ by 1, then the number of edges necessary to be removed increases by $\max(|K_i|, |K_j|) - \min(|K_i|, |K_j|)$. \square

3 Cutting plane algorithms

Cutting plane method allows us to strengthen the linear relaxation by adding inequalities. Cutting plane algorithms mainly consist in generating constraints by means of a separation procedure [see, for example, Jiinger et al. (1995), Aardal and Hoesel (1999), Vallada and Ruiz (2012) and Schrijver (2003)]. Let z^* be the incidence vector associated with the value of the variable z in the linear relaxation. The separation

problem consists in finding if there exists a valid inequality $az \leq b_0$ that cuts off the solution z^* , i.e., $az^* > b_0$. The separation algorithm associated with a family of inequalities $a^{E'}x \leq b^{E'}$ for all $E' \in \mathcal{E}'$ consists in finding a set $E' \in \mathcal{E}'$ such that $a^{E'}z^* > b^{E'}$.

The results of the previous sections have allowed us to derive some exact separation algorithms and some approximate separation algorithms. Furthermore, at the end of this section we propose a “lazy separation procedure” to ensure that the integer solutions are m -clique free interval subgraphs. In the next paragraphs, we will describe these separation algorithms for all inequalities described in the previous section.

3.1 Bipartite claw separation

In this section we describe the BC separation algorithms. We propose three algorithms to separate the bipartite claw inequalities. One is an exact algorithm and two others are heuristic procedures. We consider only $m \geq 3$ (note that it is easy to adapt the algorithm for $m = 2$).

Let the vector $z^* \in \mathbb{R}^{|\mathbb{E}|}$ be a solution of a linear relaxation. We define a weight for each edge of the complete graph G as follows: $w(e) = z_e^*$ for all $e \in E$. The separation algorithm consists in finding one bipartite claw BC such that the associated inequality is violated by z^* . This inequality is then added to the separation linear relaxation. This corresponds to find a bipartite claw $BC \subseteq E$ such that: $\sum_{e \in BC} z_e^* - \sum_{e \in \overline{BC}_\Delta} z_e^* - \sum_{e \in \overline{BC}_i} z_e^* > 5$.

From Fig. 2d, the subsets are the following:

$$\begin{aligned} BC &= \{(1, 2), (1, 3), (1, 4), (2, 5), (4, 7), (3, 6)\}. \\ BC_\Delta &= \{(2, 3), (2, 4), (3, 4)\}. \\ \overline{BC}_i &= \{(1, 5), (1, 6), (1, 7)\}. \end{aligned}$$

Proposition 17 *If a partial bipartite claw \tilde{BC} is not violated by z^* , then every bipartite claw BC including this partial bipartite claw \tilde{BC} is not violated.*

It can be observed that, from Fig. 6a if we select the 3 first vertices 1, 2 and 3 such that $(z_{1,2}^* + z_{1,3}^* - z_{2,3}^*)$ is less than 1, then there does not exist a violated bipartite claw within these 3 vertices in this position. Indeed, if $z_e^* = 1, \forall e \in BC \setminus \{(1, 2), (1, 3)\}$ and $z_{e'}^* = 0, \forall e' \in \overline{BC}_\Delta$, then we obtain in the best case, a left hand side value less than 5. Indeed, $\sum_{e \in BC \setminus \{(1,2), (1,3)\}} z_e^* = 4$. Thus, $4 + 0 + (z_{1,2}^* + z_{1,3}^* - z_{2,3}^*) < 5$. In the same way, if we add the vertex 4 and the value of $(z_{1,4}^* + z_{1,2}^* + z_{1,3}^* - z_{2,4}^* - z_{2,3}^* - z_{3,4}^*)$ is less than 2, then there does not exist a violated bipartite claw within these 4 vertices in this position. In the best case we obtain a left side value that is $3 + z_{1,4}^* + z_{1,2}^* + z_{1,3}^* - z_{2,4}^* - z_{2,3}^* - z_{3,4}^*$. Thus, the left hand side is less than 5 and the value of $\sum_{e \in BC \setminus \{(1,2), (1,3), (1,4)\}} z_e^* = 3$. With the same argument we test the weight of all partial subgraphs to drop non interesting bipartite claws. This process is used for the exact and the first heuristic algorithms.

3.1.1 Exact separation (ExBC-sep)

Now, we will explain the exact separation algorithm. The exact algorithm consists in testing all the possible seven vertices in this order (1, 2, 3, ..., 7). We select the nodes such that the values of the weighted edges in the incidence graph is maximized as follows: the weight of BC , minus the value of edges in \overline{BC}_Δ , minus the value of edges \overline{BC}_i . The running time of the exact algorithm in the worst case is bounded by $O(n^7)$. Indeed, we must select the seven vertices such that the weight of the induced graph is maximum and then the number of iterations is bounded by n^7 . Nevertheless, the running time is practically shorter with the improvement we presented before.

3.1.2 Heuristic 1: separation (HIBC-sep)

In this heuristic we start by searching the vertices 1, 2, 3, and 4 that maximize $w(1, 2) + w(1, 3) + w(1, 4) - w(2, 4) - w(2, 3) - w(3, 4)$ from Fig. 6a based on the results of Bipartite Claw in Sect. 2.1.1. If the value of $(z_{1,2}^* + z_{1,3}^* + z_{1,4}^* - z_{2,4}^* - z_{2,3}^* - z_{3,4}^*)$ is greater than 2, then it is still possible to find a violated BC inequality with this set of vertices. After that, using the greedy approach, we search to add the best weighted vertex 5 according to the incident weighted edge, if the partial BC is violated, then we will search for the best vertex 6 and with the same idea we will add vertex 7. If the BC induced by these vertices is violated by z^* , then we add the inequality to the relaxation. By using this greedy approach, the heuristic running time is $O(n^4)$.

3.1.3 Heuristic 2: separation (H2BC-sep)

This heuristic follows a greedy approach to find a violated BC inequality. We search at each step the best next vertex to add in BC . The idea is to find the 'best weighted' edge in the graph. This edge is considered as $z_{1,2}$, if the weight of the edge $z_{1,2}^* > 0$. The heuristic tries to find with the greedy approach given in Heuristic 1 the next best ordering of vertices 3, 4, ..., 7. This heuristic has $O(n^2)$ running time.

3.2 Umbrella separation

In this subsection we present three separation algorithms associated with the umbrella inequality. We propose an exact separation algorithm and two heuristics based on a greedy approach. The separation algorithm consists in finding an umbrella, such that the associated inequality is violated by z^* . If the associated inequality is violated by z^* , then we add this inequality to the linear relaxation. Thus, we search an umbrella, which is subset of E such that $\sum_{e \in E_u^a} z_e^* - \sum_{e \in E_u^i \cup E_u^c} z_e^* > 6$ where $m \geq 4$. Then, it is possible to adapt the algorithm for $m = 3$.

3.2.1 Exact separation algorithm

The exact separation algorithm starts by finding first edges $z_{1,2}$ and $z_{2,3}$. If $z_{1,2}^* + z_{2,3}^* < 1$, then the algorithm cannot find an umbrella within these two edges in this position.

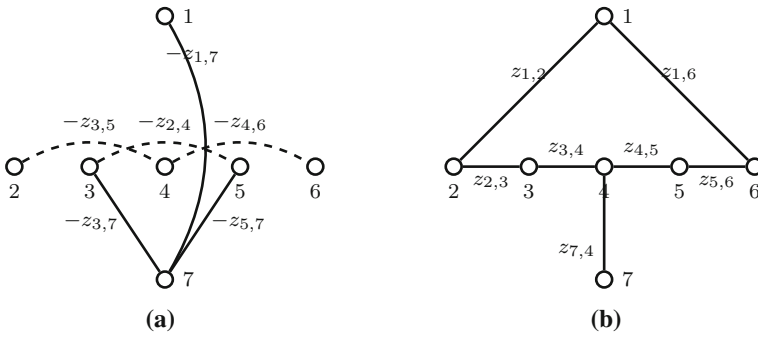
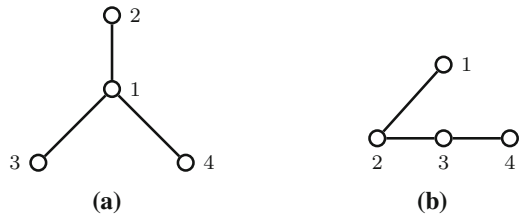


Fig. 5 Coefficient of umbrella and its complementary edges. **a** The coefficient of $E_u^t \cup E_u^c$. **b** The coefficient of E_u^a

Fig. 6 Basic edges for Bipartite Claw and Umbrella. **a** Basic nodes for BC. **b** Basic nodes for UH1



In the next step, the algorithm searches the best vertex 4, that satisfies $z_{1,2}^* + z_{2,3}^* + z_{3,4} - z_{2,4} \geq 2$, and then the algorithm adds vertex 5, then vertex 6 and finally vertex 7. Figure 5 illustrates the umbrella and the complementary subgraphs. The running time of the exact algorithm in the worst case is in $O(n^7)$.

3.2.2 HIU-sep separation

In this heuristic we start by searching the vertices 1, 2, 3, and 4 that maximize $(z_{1,2} + z_{2,3} + z_{3,4} - z_{2,4})$. In Fig. 5 the coefficient of each edge is illustrated. Figure 6b shows these basic edges. If this value is greater than 4, then we search vertex 6, and then vertex 7. If the umbrella induced by these vertices is violated by z^* , then we add this inequality. By using the greedy approach the running time is reduced to $O(n^4)$.

3.2.3 H2U-sep separation

This heuristic follows a greedy approach to find a violated umbrella inequality. We search at each step the best next edge to be added to the umbrella. The heuristic starts by the best weighted edge $z_{1,2}$, then with the greedy approach it searches the best next vertex 3 to be added to the umbrella. By following this order, the best weighted vertexes of the umbrella are determined one after one. If it is violated by z^* , then the inequality is added. This heuristic has $O(n^2)$ running time.

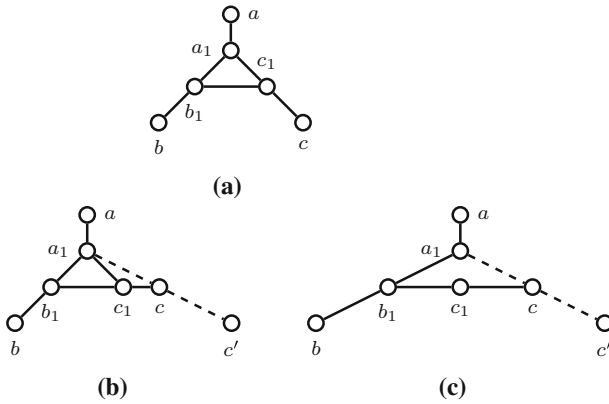


Fig. 7 Adding one vertex to 2-net to find 3-net. **a** 2-net. **b** From 2-net to 3-net. **c** 3-net

3.2.4 *n*-net separation

In this subsection we describe the *n*-net separation algorithms. Let the vector $z^* \in \mathbb{R}^{|E|}$ be a solution of the linear relaxation. The weighted vector is defined in the previous sections. The separation algorithm consists in finding one *n*-net where $n \geq 2$, such that the associated inequality is violated by z^* . This corresponds to find an *n*-net $\subseteq E$ such that $\sum_{e \in E_{net} \setminus E_{net}^h} z_e^* - \sum_{e \in \bar{E}_{net} \setminus \bar{E}_{net}^h} z_e^* > |E_{net} \setminus E_{net}^h| - 1$. From Fig. 1c, the subsets are the following:

- $E_{net} = \{(a, b), (c, 1), (d, n), (1, 2) \dots, (n - 1, n)\}$,
- $E_{net}^h = \{(b, 2), \dots, (b, n - 1)\}$.

Now, we present the *n*-net separation algorithm. Here, we search an *n*-net such that $\sum_{e \in E_{net} \setminus E_{net}^h} z_e^* - \sum_{e \in \bar{E}_{net} \setminus \bar{E}_{net}^h} z_e^* > |E_{net} \setminus E_{net}^h| - 1$.

Let us consider Fig. 7a. We search the edge (a, a_1) with the maximum weight $w((a, a_1))$. Then, we search the best vertex b_1 such that $w((a_1, b_1))$ is maximum. By a greedy search we continue for finding sequentially c_1 and c . By considering 2-net, if the value of $\sum_{e \in E_{net} \setminus E_{net}^h} z_e^* - \sum_{e \in \bar{E}_{net} \setminus \bar{E}_{net}^h} z_e^*$ is greater than 6, then we try to extend it to a 3-net in the following way: By adding a vertex (c') adjacent to c . Figure 7b explains the process, when the dashed edges are added.

It is important to mention that we remove the edge (a_1, c_1) (see. Fig. 7c) if $n + 1$ net is violated. Then, we add the *n*-net inequality (if $n \geq 2$) and we stop the algorithm.

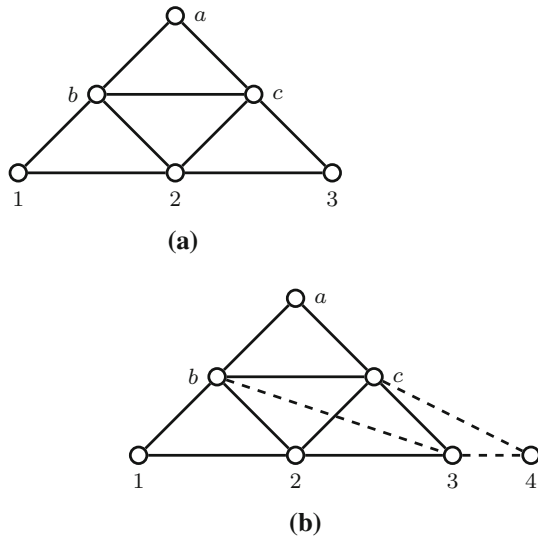
The proposed heuristic running time is $O(n^2)$.

3.2.5 *n*-tent separation

For the *n*-tent separation algorithm we propose the following algorithm, which is based on a greedy approach.

Again, the vector $z^* \in \mathbb{R}^{|E|}$. The *n*-tent separation algorithm consists in finding an *n*-tent, such that the associated inequality is violated by z^* . Thus, we search an

Fig. 8 Adding one vertex to 3-tent to find 4-tent. **a** 3-nent. **b** 4-tent



n -tent where $n \geq 3$, n -tent $\subseteq E$ such that $\sum_{e \in E_{tent} \setminus E_{tent}^h} z_e - \sum_{e \in \bar{E}_{tent} \setminus E_{tent}^{\bar{h}}} z_e \leq |E_{tent} \setminus E_{tent}^h| - 1$.

We search for the best $z_{a,b}$, and $z_{a,c}$ in the first step (see 8a). In the next step, we search the best nodes 1, 2, 3 one by one using a greedy approach to maximize the sum of n -tent edges, by considering the weight of the edges from n -tent and its complementary. If 2-tent is not found, this means that 3-tent does not exist. If 3-tent is found, then we try to add the best node 4, which is connected with node c' and connecting c' to a_1 and to remove (a_1, c) to have the structure of 4-tent. The heuristic continues for searching for $n + 1$ -net at each step. If the heuristic failed to find $n + 1$ -net, then the violated n -net inequality associated with the found n -net will be added. The proposed heuristic running time is $O(n^3)$.

3.3 Hole separation

Now, we explain the hole separation algorithm, which is based on a greedy approach. As in the previous paragraphs, the relaxed solution is represented by vector $z^* \in \mathbb{R}^{|E|}$. The separation algorithm consists in finding a forbidden subgraph hole, such that the associated hole inequality is violated by z^* . Thus, we search a hole of size C where $holeC \subseteq E$ such that $\sum_{e \in C} (|C| - 3)z_e - \sum_{e \in \bar{C}} z_e > (|C| - 1)(|C| - 3)$.

The algorithm starts by the edge (u_1, u_2) with $w((u_1, u_2))$ is maximum (see Fig. 9). By a greedy search, we pick the vertex u_3 such that $w((u_2, u_3)) - w((u_1, u_3))$ is maximum. With the same process, we search a sequence of vertices u_4, \dots, u_n . In each step, we consider the cycle where we connect u_1 to u_n . We find a hole where the associated inequality is violated and then we add it.

Fig. 9 Hole

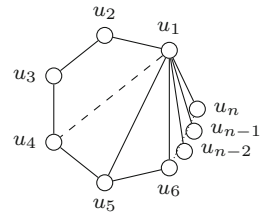
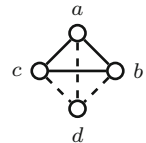


Fig. 10 Clique of size 4



If $\sum_{e \in C} (|C| - 3)z_e - \sum_{e \in \bar{C}} z_e \leq (|C| - 1)(|C| - 3)$ where C is the incident cycle, then we stop the algorithm since we cannot find a cycle where the associated inequality is violated.

3.4 Clique separation

In this subsection we explain the clique separation algorithm. The vector $z^* \in \mathbb{R}^{|E|}$ represents the solution. The clique separation algorithm consists in finding a clique, such that the associated inequality is violated by z^* . More precisely, we search a clique of size K , $E(K) \subseteq E$ such that $\sum_{e \in E(K)} z_e > |E(K)| - f(K, m)$.

The heuristic starts by the edge (a, b) such that $w((a, b))$ is maximum. By the greedy search we find the vertex c such that $w((a, c)) + w((a, b)) + w((c, b))$ is maximum. By the same process we search the vertex d such that $w((a, c)) + w((a, b)) + w((c, b)) + w((b, d)) + w((c, d)) + w((a, d))$ is maximum (see Fig. 10). In each step we consider the clique where we connect a vertex with all other vertices in the clique. If we find a clique where the associated inequality is violated, then this inequality is added.

If $\sum_{e \in E(K)} z_e \leq |E(K)| - f(K, m)$, then we stop the algorithm, since we cannot find a clique where associated inequality is violated, and we add the inequality of clique of size $n - 1$.

3.5 Lazy constraint approach

In this subsection, we propose some algorithms to ensure that a solution given by an integer value vector $\bar{z} \in \{0, 1\}^{|V| \times |V|}$ induces an m -clique free interval subgraph. We consider the induced graph $\tilde{G} = (V, \tilde{E})$ where \tilde{E} contains all edges such that $\bar{z}_e = 1$. For the interval graph detection, we use the algorithm given in Habib et al. (2000) to check if \tilde{G} is an interval graph. If \tilde{G} is not an interval graph, then we add an interdiction inequality

$$\sum_{e \in \tilde{E}} z_e - \sum_{e \in E \setminus \tilde{E}} z_e \leq |\tilde{E}| - 1.$$

This algorithm runs in $O(n + m \log(n))$ Habib et al. (2000).

For the clique inequalities, we search the clique of a maximum size. We use the integer linear program given in Pardalos and Xue (1994) to find the maximum clique in \tilde{G} . Finding the maximum clique in G is an NP-hard problem. We use CPLEX to solve this problem. If \tilde{G} contains a clique K of a size greater than m , then we add the following clique inequality associated with K .

$$\sum_{e \in E(K)} z_e \leq |E(K)| - 1.$$

We can note that, the lazy constraints are used only to check the validity of an integer solution.

If no clique inequality and no interdiction inequality are generated, then the solution is feasible.

4 Application to the generalized open shop problem with disjunctive constraints

The Generalized Open Shop with Disjunctive Constraints (GOSDC) can be formulated as follows. Let M be the set of machines. For every $i \in M$ we consider the set of jobs J_i to be performed on machine i and we denote by $\mathcal{J} = \{J_1, \dots, J_m\}$ the set of all these sets and by $J = \bigcup_{i \in M} J_i$ the union of these sets. We denote by p_{ij} the processing time of job j on its machine i . We consider an incompatibility graph $G_I = (V_I, E_I)$ such that for each job $j \in J$ we associate a vertex $v_j \in V_I$ and there exists an edge between v_{j_1} and v_{j_2} if j_1 and j_2 cannot run at the same time. It is worthnoting that it is necessary to consider a linear ordering on each machine. To the best of our knowledge, this special problem has not been studied before.

4.1 Integer linear programming formulation

In this subsection, we present an integer linear programming (ILP) formulation for solving the problem. We need a family of binary variables. In the following, we describe the variables used in the formulation:

$$\bar{z}_{j_1, j_2} = \begin{cases} 1 & \text{if job } j_1 \text{ runs before job } j_2 \\ 0 & \text{otherwise} \end{cases} \quad \forall j_1, j_2 \in J.$$

$$z_{j_1, j_2} = \begin{cases} 1 & \text{if } j_1 \text{ and } j_2 \text{ run at the same time} \\ 0 & \text{otherwise} \end{cases} \quad \forall j_1 \in J_i, \quad j_2 \in J_{i'} | i \neq i' \in M.$$

For every $j \in J$, we consider the variable $y_j \in \mathbb{N}^+$ representing the starting time of job j .

$C_{\max} \in \mathbb{N}^+$ is the maximum completion time, such that $C_{\max} \leq C$, where C is an upper bound obtained heuristically.

The GOSDC can be solved by the following ILP, denoted by (P_{GOS}):

$$\begin{aligned} \min C_{max} \\ y_j + p_{ij} \leq C_{max}, \quad \forall i \in M \quad \forall j \in J_i, \end{aligned} \tag{13}$$

$$y_{j_1} + p_{ij_1} \leq y_{j_2} + C \cdot \bar{z}_{j_2, j_1}, \quad \forall i \in M \quad \forall j_1 \in J_i \text{ and } j_2 \in J, \tag{14}$$

$$\bar{z}_{j_1, j_2} + \bar{z}_{j_2, j_1} = 1, \quad \forall i \in M \quad \forall j_1, j_2 \in J_i, \tag{15}$$

$$\bar{z}_{j_1, j_2} + \bar{z}_{j_2, j_1} = 1, \quad \forall (v_{j_1}, v_{j_2}) \in E_I, \tag{16}$$

$$\bar{z}_{j_1, j_2} + \bar{z}_{j_2, j_1} + z_{j_2, j_1} = 1, \quad \forall (v_{j_1}, v_{j_2}) \notin E_I, \tag{17}$$

$$\sum_{(j_1, j_2) \in E(\bar{I})} z_{j_1, j_2} - \sum_{(j_1, j_2) \in E \setminus E(\bar{I})} z_{j_1, j_2} \leq |E(\bar{I})| - 1, \quad \forall \bar{I} \subseteq \bar{\mathcal{I}}, \tag{18}$$

$$\sum_{(j_1, j_2) \in E(K)} z_{j_1, j_2} \leq |E(K)| - 1, \quad \forall K \subseteq \mathcal{K}, \tag{19}$$

$$C_{max} \in \mathbb{N}^+, \tag{20}$$

$$y_j \in \mathbb{N}^+ \quad \forall i \in M \quad \forall j \in J_i, \tag{21}$$

$$z_{j_1, j_2} \in \{0, 1\} \quad \forall i \in M \quad \forall j_1, j_2 \in J_i, \tag{22}$$

$$\bar{z}_{j_2, j_1} \in \{0, 1\} \quad \forall i \in M \quad \forall j_1, j_2 \in J_i, \tag{23}$$

The objective function is to minimize the makespan. Inequalities (13) ensure that the starting time for each job plus its processing time is less than or equal to the total completion time. Inequalities (14) and inequalities (15) guarantee that there is no two jobs running on the same machine at the same time and control the linear ordering. Inequalities (16) ensure that if two jobs are linked by an edge in the compatibility graph, then they do not run at the same time. Indeed, for these two jobs j_1 and j_2 either j_1 before j_2 or j_2 before j_1 . Inequalities (17) ensure the three possibilities: j_1 before j_2 or j_2 before j_1 or they run at the same time. Inequalities (18) and (19) guarantee that the induced subgraphs are m -clique free interval subgraphs. The number of inequalities may be exponential and thus we will use the separation algorithm presented in the previous section.

In the proposed formulation, $\bar{\mathcal{I}}$ is the set of all subgraphs such that for all $\bar{I} \in \bar{\mathcal{I}}$, $G[\bar{I}]$ is not an interval subgraph. On the other hand, the set \mathcal{K} contains all the cliques of size $m + 1$ included in V_I . It is worth noting that the proposed valid inequalities in the previous part of this paper will ensure that the ILP will restrict the exploration on a reduced feasible space since every feasible schedule of P_{GOS} can be associated with an m -clique free interval subgraph defined on G_I . In other words, the projection on the variables $z_{i,j}$ gives the polytope we previously studied. The constraints (18) and (19) corresponds to m -clique free interval subgraphs.

5 Experimental results

In order to evaluate the efficiency of the inequalities mentioned in Sect. 2.1, we developed the mentioned exact and heuristic separations. All computational results are obtained using Cplex 12.6 and JAVA for implementing exact and heuristic algorithms.

The ILP with the valid inequalities is tested on the following proposed benchmark of instances.

The processing times are uniformly distributed between 50 and 150 as it is common in the literature Hall and Posner (2001). The graph density (GD) is equal to 0.5 and calculated as follows: $GD = \frac{|E|}{|V|(|V|-1)}$ where E is the set of edges associated with precedence constraints between jobs and V is the set of vertices associated with jobs. The results are given for 4 families of instances. Each family contains 5 instances with the same parameters.

The numerical tests have been carried out on an Intel Core i5 of 3.4 GHz on linux environment. The required CPU time is measured in seconds. We limit to 3600 s the algorithm running time for each instance, by using 4.0 GB of RAM.

The next tables provide the following information:

- $|J_i|$: Number of jobs per machine.
- m : Number of machines.
- $|J|$: Number of jobs.
- rootGap : The gap between the lower bounds and the upper bounds ($100 \times \frac{UB-LB}{LB}$) at the the root node.
- Method: Inequalities used for the separation:
 - 0: Basic model;
 - 1: Only the bipartite claw inequalities (H1BC-Sep),
 - 2: Only the Umbrella inequalities (H1U-sep),
 - 3: Only the Hole inequalities,
 - 4: Only the Clique–hole inequalities,
 - 5: Only the n -net inequalities,
 - 6: Only the n -tent inequalities,
 - 7: All inequalities of methods 1 to 6,
 - 8: Cplex with only its generic cuts
- Nodes : The number of nodes in the branching tree.
- gap : The gap between the lower bounds and the upper bounds ($100 \times \frac{UB-LB}{LB}$).
- CPU: Computational time (limited to 1 h).
- Ct BC: The number of bipartite claw inequalities added in the Branch-and-cut method (B&C).
- Ct UMB: The number of umbrella inequalities added in the B&C.
- Ct H: The number of hole inequalities added in the B&C.
- Ct Q: The number of clique–hole inequalities added in the B&C.
- Ct NN: The number of n -net inequalities added in the B&C.
- Ct NT: The number of n -tent inequalities added in the B&C.
- o/p: The number of solved instances among the 5 tested instances.

For the solving methods numbered 0 to 7 we remove all the proper improvements of CPLEX.

Notice that when $m = 2$ we do not test the options 2 and 3 since the hole inequalities, clique inequalities and umbrella inequalities are dominated by or equivalent to the inequalities (9). For the option 4 and $m = 2$ we consider an adaptation of the clique separation proposed in Sect. 3.4 where right-hand-side is equal to the number of vertices in the current clique. We present the results where we already selected the

Table 1 Average CPU time

$ J_i $	m	$ J $	rootGap	Method	CPU	Nodes	Gap	o/p
5	2	10	0.44	0	0	248.2	0.00	5/5
5	2	10	0.44	1	0	384.6	0.00	5/5
5	2	10	0.44	4	0	152.6	0.00	5/5
5	2	10	0.44	5	0	248.2	0.00	5/5
5	2	10	0.44	6	0	232.8	0.00	5/5
5	2	10	0.44	7	0	143.4	0.00	5/5
5	2	10	0.44	8	0	272.2	0.00	5/5
10	2	20	0.48	0	2182.6	888005.6	0.05	2/5
10	2	20	0.48	1	3600	1384352	0.07	0/5
10	2	20	0.48	4	930.2	385605.2	0.00	4/5
10	2	20	0.48	5	2182.6	867387.4	0.04	2/5
10	2	20	0.48	6	1564.6	649562.6	0.03	3/5
10	2	20	0.48	7	1579.2	554582.4	0.01	3/5
10	2	20	0.48	8	452.8	214562	0.00	5/5
5	4	20	0.66	0	3600	283161	0.59	0/5
5	4	20	0.66	1	3600	252580	0.48	0/5
5	4	20	0.66	2	3600	132601	0.57	0/5
5	4	20	0.66	3	3600	168232.4	0.46	0/5
5	4	20	0.66	4	3600	213900.4	0.57	0/5
5	4	20	0.66	5	3600	186662	0.58	0/5
5	4	20	0.66	6	3600	137594.8	0.59	0/5
5	4	20	0.66	7	3600	353032.8	0.27	0/5
5	4	20	0.66	8	3600	260540.2	0.59	0/5
5	6	30	0.78	0	3600	209214.4	0.77	0/5
5	6	30	0.78	1	3600	18177.8	0.77	0/5
5	6	30	0.78	2	3600	20070.2	0.77	0/5
5	6	30	0.78	3	3600	86451.6	0.77	0/5
5	6	30	0.78	4	3600	138560.2	0.77	0/5
5	6	30	0.78	5	3600	146006.8	0.77	0/5
5	6	30	0.78	6	3600	92747.4	0.77	0/5
5	6	30	0.78	7	3600	30685.6	0.77	0/5
5	6	30	0.78	8	3600	127214.2	0.77	0/5

best separation algorithm for each family of inequalities i.e., the H1BC-sep for the *BC* inequalities and the H1U-sep for the umbrella inequalities. Table 1 presents the results for seven different methods. First, we observe that for all instances on 2 machines CPLEX with its proper improvements can solve all of them to optimality. Furthermore, with 5 jobs per machine the number of nodes decreases when we add cuts . In this case, we notice that adding all the inequalities is the best option in order to reduce the number of generated nodes. With 2 machines and 10 jobs per machine, if we do not

consider the method 8, we notice that we reduce the CPU time and the option 4 is the most efficient for solving this family of instances. Indeed, the CPU time is only 930 seconds with the method 4. Secondly, if we increase the number of machines, then we cannot solve instances with 5 jobs per machine. However, we can reduce the gap by using our valid inequalities. For 5 jobs per machine and 4 machines, the gap is divided by 2 by using all the inequalities, whereas for 6 machines all methods have the same gap. We can note that the rootGap is not always dependent of the solving method. Two reasons can explain this observation: either it is due to the value of C or it is due to the added inequalities, which focus on only one family of variables.

Table 2 presents a complement of informations with the details of the average of the number of generated inequalities during the Branch-and-cut procedure. We remark that, when $m > 2$, we generate a small number of clique inequalities.

6 Conclusion and perspectives

In this paper we considered the m -clique free interval subgraphs. We studied the facial structure of their polytope and we proposed some facet-defining inequalities to reinforce the associated linear relaxation. Moreover, the GOSDC was studied as an application in scheduling. Exact and heuristic separation algorithms were presented and used into a Branch-and-cut procedure for solving the GOSDC. Computational experiments were carried out and the results show that the algorithms are capable to solve many instances to optimality within reasonable computation time.

As a perspective, further research in this direction will be helpful to strengthen the integer programming formulations of a large variety of scheduling problems as another application to the m -clique free interval subgraphs.

Appendix

Proof of Proposition 2 Let us denote by $az \leq \alpha$ inequality $-z_e \leq 0$ associated with $e \in E$. Let $bz \leq \beta$ be a facet defining inequality of $P_{\mathcal{I}}(G, m)$, such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Clearly $I_0 = \emptyset$ is feasible. Let $I_0 = \emptyset$ be a solution, and the associated incidence vector z^{I_0} verifies $-z_e = 0$.

Let $e' \in E \setminus \{e\}$. The solution $I_{e'} = \{e'\}$ is feasible and the incidence vector $z^{I_{e'}}$ associated with $I_{e'}$ verifies $-z_e = 0$.

Since, $az^{I_0} = az^{I_{e'}}$, we deduce that $bz^{I_0} = bz^{I_{e'}}$. This implies that $b(e') = 0$ for all $e' \in E \setminus \{e\}$. Therefore, we set $b(e) = \rho$, and then $b = \rho a$. \square

Proof of Proposition 3 Let us denote by $az \leq \alpha$ inequality $z_e \leq 1$ associated with $e \in E$. Let $bz \leq \beta$ be a facet defining inequality of $P_{\mathcal{I}}(G, m)$, such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Consider the feasible solution $I_0 = \{e\}$. And the associated incidence vector z^{I_0} verifies $z_e = 1$.

Table 2 Average number of inequalities

$ J_i $	m	Method	Ct BC	Ct UMB	Ct H	Ct Q	Ct NN	Ct NT
5	2	0	0	0	0	0	0	0
5	2	1	3.6	0	0	0	0	0
5	2	4	0	0	0	10.4	0	0
5	2	5	0	0	0	0	0	0
5	2	6	0	0	0	0	0	1
5	2	7	2	0	0	5.4	0	0.8
5	2	8	0	0	0	0	0	0
10	2	0	0	0	0	0	0	0
10	2	1	1210.2	0	0	0	0	0
10	2	4	0	0	0	1157	0	0
10	2	5	0	0	0	0	1.6	0
10	2	6	0	0	0	0	0	92.6
10	2	7	549.6	0	0	822.6	0	44.2
10	2	8	0	0	0	0	0	0
5	4	0	0	0	0	0	0	0
5	4	1	7654.2	0	0	0	0	0
5	4	2	0	6962.4	0	0	0	0
5	4	3	0	0	4182.8	0	0	0
5	4	4	0	0	0	8	0	0
5	4	5	0	0	0	0	288	0
5	4	6	0	0	0	0	0	1781.4
5	4	7	5966.6	2207.2	2043.4	8.2	591.6	1189
5	4	8	0	0	0	0	0	0
5	6	0	0	0	0	0	0	0
5	6	1	28265	0	0	0	0	0
5	6	2	0	28241.6	0	0	0	0
5	6	3	0	0	6781.8	0	0	0
5	6	4	0	0	0	132.8	0	0
5	6	5	0	0	0	0	524.2	0
5	6	6	0	0	0	0	0	9771.4
5	6	7	11780.8	10271.8	1316.6	96.8	563.6	4493.4
5	6	8	0	0	0	0	0	0

Let $e' \in E \setminus \{e\}$. The solution $I_{e'} = \{e', e\}$ is feasible and the incidence vector $z^{I_{e'}}$ associated with $I_{e'}$ verifies $z_e = 1$.

Since, $az^{I_0} = az^{I_{e'}}$, we deduce that $bz^{I_0} = bz^{I_{e'}}$. This implies that $b(e') = 0$ for all $e' \in E \setminus \{e\}$. Therefore, we set $b(e) = \rho$, and then $b = \rho a$. □

Proof of Proposition 4 Let us denote by $az \leq \alpha$ the inequality (1). Let $bz \leq \beta$ be an inequality that defines a facet of $P_I(G, m)$, such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq$

$\{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. Since $P_1(G, m)$ is full dimensional, we need to prove that there exists ρ such that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $e_1, e_2 \in BC$ be two edges. Clearly, the solutions $I_1 = BC \setminus \{e_1\}$, and $I_2 = BC \setminus \{e_2\}$ are feasible. Their incidence vectors satisfy the inequality (1) with equality. Since, $az^{I_1} = az^{I_2}, bz^{I_1} = bz^{I_2}$, it follows that $b(e_1) = b(e_2)$. We set $b(e_1) = \rho$. As e_1 and e_2 are arbitrary in BC , then $b(e) = b(e')$ for all $e, e' \in BC$.

The solutions $I_3 = BC \setminus \{(1, 4)\}$, and $I_4 = I_3 \cup \{(2, 4)\}, I_5 = I_3 \cup \{(5, 4)\}$, and $I_6 = I_3 \cup \{(5, 7)\}$ are feasible. Their incidence vectors satisfy the inequality (1) with equality. Since, $az^{I_3} = az^{I_4} = az^{I_5} = az^{I_6}, bz^{I_3} = bz^{I_4} = bz^{I_5} = bz^{I_6}$, it follows that $b((2, 4)) = b((5, 4)) = b((5, 7)) = 0$. By symmetry, $b(e) = 0$, for all $e \in \overline{BC}_h^4 \cup \overline{BC}_h^5 \cup \overline{BC}_\Delta$.

The solutions $I_7 = BC \setminus \{(4, 7)\}$ and $I_8 = I_7 \cup \{(1, 7)\}$ are feasible and verify the inequality (1) with equality. Since, $az^{I_7} = az^{I_8}, bz^{I_7} = bz^{I_8}$, it follows that $b((1, 7)) = 0$. By symmetry, we have $b((1, 5)) = b((1, 6)) = 0$.

Let $e_3 \in E \setminus (BC \cup \overline{BC})$. The solutions I_3 and $I_9 = I_3 \cup \{e_3\}$ are feasible and verify the inequality (1) with equality. Since, $az^{I_3} = az^{I_9}, bz^{I_3} = bz^{I_9}$, which implies that $b(e_3) = 0$. By symmetry, we have $b(e) = 0$ for all $e \in E \setminus (BC \cup \overline{BC})$. Thus $b = \rho a$. □

Proof of Proposition 6 Let us denote by $az \leq \alpha$ the inequality (3) associated with e . Let $bz \leq \beta$ be a facet defining inequality of $P_1(G, m)$ such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $e_1, e_2 \in E_u^a \setminus \{(4, 7)\}$. The solutions $I_1 = E_u \setminus \{e_1\}$ and $I_2 = E_u \setminus \{e_2\}$ are feasible and the incidence vectors z^{I_1} and z^{I_2} verify the inequality (3) with equality. Moreover, we have $az^{I_1} = az^{I_2}$. Hence, $bz^{I_1} = bz^{I_2}$. This implies that $b(e_1) = b(e_2)$. We set $b(e_1) = \rho$. As e_1 and e_2 are arbitrary in $E_u^a \setminus \{(4, 7)\}$, by symmetry, we have $b(e') = b(e) = \rho$ for all $e \in E_u^a \setminus \{(4, 7)\}$.

The solution $I_3 = E_u^a \setminus \{(4, 7), (1, 6)\}$ and $I_4 = (I_3 \cup \{(2, 5)\}) \setminus \{(4, 5)\}$ are feasible and the incidence vectors z^{I_3} and z^{I_4} verify the inequality (3) with equality. Moreover, we have $az^{I_3} = az^{I_4}$. Hence, $bz^{I_3} = bz^{I_4}$. This implies that $b((4, 5)) = b((2, 5))$. Thus, from the previous results we deduce that $b((2, 5)) = b(e)$ for all $e \in E_u^a \setminus \{(4, 7)\}$.

The solution $I_5 = (I_3 \cup \{(2, 6)\}) \setminus \{(5, 6)\}$ is feasible, such that the incidence vectors z^{I_5} and z^{I_3} verify the inequality (3) with equality. Moreover, we have $az^{I_5} = az^{I_3}$. Hence, $bz^{I_5} = bz^{I_3}$. This implies that $b((2, 6)) = b((5, 6))$. Indeed, from the previous results we have $b((2, 6)) = b(e)$ for all $e \in E_u^a \setminus \{(4, 7)\}$.

The solution $I_6 = (I_3 \cup \{(3, 6)\}) \setminus \{(5, 6)\}$ is feasible and the incidence vector z^{I_6} and z^{I_3} verify the inequality (3) with equality. Moreover, we have $az^{I_6} = az^{I_3}$. Hence, $bz^{I_6} = bz^{I_3}$. This implies that $b((3, 6)) = b((5, 6))$. Indeed, from the previous results we can deduce that $b((3, 6)) = b(e)$ for all $e \in E_u^a \setminus \{(4, 7)\}$.

The solutions $I_7 = E_u^a \setminus \{(4, 7) \cup (1, 6)\}$ and $I_8 = (I_7 \cup \{(4, 7)\}) \cup E_u^i$ are feasible and the incidence vectors z^{I_7} , and z^{I_8} verify the inequality (3) with equality. Moreover, we have $az^{I_7} = az^{I_8}$. Hence $bz^{I_7} = bz^{I_8}$. This implies that $b(e) = 0$ for all $e \in E_u^i$.

Let $e \in \{(2, 7), (3, 7), (1, 7), (5, 7), (6, 7)\}$. The solutions I_1 and $I_9 = I_1 \cup \{e\}$ are feasible and the incidence vectors z^{I_1}, z^{I_9} verify the inequality (3) with equality.

Moreover, we have $az^{I_1} = az^{I_9}$. Hence, $bz^{I_1} = bz^{I_9}$. This implies that $b((2, 7)) = b((3, 7)) = b((1, 7)) = b((5, 7)) = b((6, 7)) = 0$. Thus $b = \rho a$. \square

Proof of Proposition 7 Let us denote by $az \leq \alpha$ inequality (4) associated with e . Let $bz \leq \beta$ be a facet defining inequality of $P_I(G, m)$ such that $\{z \in P_I(G, m) : az = \alpha\} \subseteq \{z \in P_I(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $e_1, e_2 \in E_u^t \cup E_u^c$ be two edges, where $e_1 \neq e_2$. We consider the edge sets $I_1 = E_u \cup \{e_1\}$ and $I_2 = E_u \cup \{e_2\}$ where the incidence vectors z^{I_1} and z^{I_2} are solutions of $P_I(G, m)$ and satisfy the inequality (4) with equality. Moreover, we have $az^{I_1} = az^{I_2}$. Thus, $bz^{I_1} = bz^{I_2}$. This implies that $b(e_1) = b(e_2)$. As e_1, e_2 are arbitrary, then $b(e) = b(e')$ for all $e, e' \in E_u^t \cup E_u^c$.

Let $e_3 \in E_u^a$. The solution $I_3 = E_u \setminus \{e_3\}$ is feasible and the incidence vector z^{I_3} verifies the inequality (4) with equality. Moreover, we have $az^{I_1} = az^{I_3}$. Hence, $bz^{I_1} = bz^{I_3}$. This implies that $b(e_3) = -b(e_1)$. Thus, by symmetry we have $b(e') = -b(e)$ for all $e \in E_u^t \cup E_u^c, e' \in E_u^a$.

The solutions $I_4 = E_u \setminus \{(1, 6)\}, I_5 = (E_u \setminus \{(1, 6)\}) \cup E_u^i, I_6 = I_5 \setminus \{(1, 5)\}, I_7 = I_6 \setminus \{(1, 4)\}$ and $I_8 = I_7 \setminus \{(1, 3)\}$ are feasible. The incidence vectors $z^{I_4}, z^{I_5}, z^{I_6}, z^{I_7}$ and z^{I_8} verify the inequality (3) with equality. Since $az^{I_4} = az^{I_5} = az^{I_6} = az^{I_7} = az^{I_8}, bz^{I_4} = bz^{I_5} = bz^{I_6} = bz^{I_7} = bz^{I_8}$, it follows that $b(e) = 0$ for all $e \in E_u^i$.

The solutions $I_9 = E_u \cup \{(2, 7), (2, 4)\}, I_{10} = E_u \cup \{(2, 5), (2, 4)\}, I_{11} = E_u \cup \{(2, 4)\}, I_{12} = E_u \setminus \{(2, 3)\}, I_{13} = I_{12} \cup \{(2, 6)\}, I_{14} = I_{12} \cup \{(2, 5)\}, I_{15} = E_u \setminus \{(5, 6)\}$ and $I_{16} = I_{15} \cup \{(3, 6)\}$ are feasible and verify the inequality (4) with equality. Since $az^{I_9} = az^{I_{10}} = az^{I_{11}} = az^{I_{12}} = az^{I_{13}} = az^{I_{14}} = az^{I_{15}} = az^{I_{16}}, bz^{I_9} = bz^{I_{10}} = bz^{I_{11}} = bz^{I_{12}} = bz^{I_{13}} = bz^{I_{14}} = bz^{I_{15}} = bz^{I_{16}}$, it follows that $b((2, 6)) = b((2, 5)) = b((2, 7)) = b((6, 7)) = b((3, 6)) = 0$.

Let $e \in E \setminus (E_u \cup \bar{E}_u)$. The solutions $I_{17} = E_u \setminus \{(4, 7)\}$ and $I_{18} = I_{17} \cup \{e\}$ are feasible and verify the inequality (4) with equality. Since $az^{I_{17}} = az^{I_{18}}, bz^{I_{17}} = bz^{I_{18}}$. By symmetry, we have $b(e) = 0$.

We set $b(e) = \rho$ for $e \in E_u$ and the proof is ended. \square

Proof of Proposition 8 Let us denote by $az \leq \alpha$ the inequality (5) associated with e . Let $bz \leq \beta$ be a facet defining inequality of $P_I(G, m)$ such that $\{z \in P_I(G, m) : az = \alpha\} \subseteq \{z \in P_I(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $e_1, e_2 \in E_{net} \setminus E_{net}^h$ be two edges, where $e_1 \neq e_2$. We consider the edge sets $I_1 = E_{net} \setminus \{e_1\}$ and $I_2 = E_{net} \setminus \{e_2\}$. Their incidence vectors z^{I_1} and z^{I_2} are solutions of $P_I(G, m)$ and satisfy the inequality (5) with equality. Moreover, we have $az^{I_1} = az^{I_2}$ and then $bz^{I_1} = bz^{I_2}$. This implies that $b(e_1) = b(e_2)$. As e_1, e_2 are arbitrary, then $b(e) = b(e')$ for all $e, e' \in E_{net} \setminus E_{net}^h$.

We consider the edge sets $I_3 = E_{net} \setminus \{(b, 1)\}$ and $I_4 = E_{net} \setminus (\{(b, 1)\} \cup E_{net}^h)$. Their incidence vectors z^{I_3} and z^{I_4} are solutions of $P_I(G, m)$ and satisfy the inequality (5) with equality. Moreover, we have $az^{I_3} = az^{I_4}$. Hence, we have $bz^{I_3} = bz^{I_4}$. This implies that $b(e) = 0$ for all $e \in E_{net}^h$.

Let $e_3 \in \bar{E}_{net} \setminus \bar{E}_{net}^h$. The solution $I_5 = E_{net} \cup \{e\}$ is feasible and satisfies the inequality (5) with equality. Moreover, we have $az^{I_1} = az^{I_5}$ and then $bz^{I_1} = bz^{I_5}$. This implies that $b(e) = -b(e')$ for all $e \in E_{net} \setminus E_{net}^h$ and $e' \in \bar{E}_{net} \setminus \bar{E}_{net}^h$.

Considering the edge sets $I_6 = E^{net} \cup \{(c, 2), (c, 3), \dots, (c, n), (c, d)\}$, $I_7 = E^{net} \cup \{(c, 2)\}$, $I_8 = E^{net} \cup \{(d, 1), (d, 2), \dots, (d, n - 1)\}$, $I_9 = E^{net} \cup \{(d, n - 1)\}$, $I_{10} = E^{net} \cup \{(a, 1)\}$, $I_{11} = E^{net} \cup \{(a, 1), (a, c)\}$, $I_{12} = E^{net} \cup \{(a, n)\}$, $I_{13} = E^{net} \cup \{(a, n), (a, d)\}$. These edge sets are solutions and satisfy the inequality (5) with equality. Moreover, we have $az^{I_6} = az^{I_7} = az^{I_8} = az^{I_9} = az^{I_{10}} = az^{I_{11}} = az^{I_{12}} = az^{I_{13}}$. Hence, we have $bz^{I_6} = bz^{I_7} = bz^{I_8} = bz^{I_9} = bz^{I_{10}} = bz^{I_{11}} = bz^{I_{12}} = bz^{I_{13}}$. Thus, $b(e) = 0$ for all $e \in E_{net}^h$.

Let $e \in E \setminus (E_{net} \cup \overline{E_{net}})$. By considering the feasible solutions $I_{14} = E_{net} \setminus \{(a, b)\}$ and $I_{15} = E_{net} \setminus \{(a, b)\} \cup \{e\}$, we have $az^{I_{14}} = az^{I_{15}}$. Hence, $bz^{I_{14}} = bz^{I_{15}}$. This implies that $b(e) = 0$.

We set $b(e) = \rho$ for $e \in E_{net} \setminus E_{net}^h$ and the proof is complete. □

Proof of Proposition 9 Let us denote by $az \leq \alpha$ the inequality (6) associated with e . Let $bz \leq \beta$ be a facet defining inequality of $P_I(G, m)$ such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $e_1, e_2 \in E_{tent} \setminus E_{tent}^h$ be two edges, where $e_1 \neq e_2$. We consider the edge sets $I_1 = E_{tent} \setminus \{e_1\}$ and $I_2 = E^{tent} \setminus \{e_2\}$. Their incidence vectors z^{I_1} and z^{I_2} are solutions of $P_{\mathcal{I}}(G, m)$ and satisfy the inequality (6) with equality. Moreover, we have $az^{I_1} = az^{I_2}$ and then $bz^{I_1} = bz^{I_2}$. This implies that $b(e_1) = b(e_2)$. As e_1, e_2 are arbitrary, then $b(e) = b(e')$ for all $e, e' \in E_{tent} \setminus E_{tent}^h$.

We consider the edge sets $I_3 = E_{tent} \setminus \{(a, c)\}$, and $I_4 = E^{tent} \setminus (\{(a, c)\} \cup \{(b, c)\})$ there incidence vectors z^{I_3} , and z^{I_4} are solutions of $P_{\mathcal{I}}(G, m)$ and satisfy the inequality (6) with equality. Moreover, we have $az^{I_3} = az^{I_4}$ and then $bz^{I_3} = bz^{I_4}$. This implies by symmetry that $b((b, c)) = b((b, 2)) = b((c, 4)) = 0$.

Let $e_3 \in \overline{E_{tent}} \setminus E_{tent}^h$. The solution $I_5 = E^{tent} \cup \{e\}$ is feasible and satisfies the inequality (6) with equality. Moreover, we have $az^{I_1} = az^{I_5}$. Hence, $bz^{I_1} = bz^{I_5}$. This implies that $b(e) = -b(e')$ for all $e \in E_{tent} \setminus E_{tent}^h$ and $e' \in \overline{E_{tent}} \setminus E_{tent}^h$.

Let $(i, i + 3) \in E_{tent}^h$. The solutions $I_6 = E^{tent} \cup \{(i, i + 3), (i, i + 2)\}$ and $I_7 = E^{tent} \cup \{(i, i + 2)\}$ are feasible and satisfy the inequality (6) with equality. Moreover, we have $az^{I_6} = az^{I_7}$ and then $bz^{I_6} = bz^{I_7}$. Thus, $b(e) = 0$ for all $e \in E_{tent}^h$.

Let $e \in E \setminus (E_{tent} \cup \overline{E_{tent}})$. By considering the feasible solutions $I_8 = E_{tent} \setminus \{(a, c)\}$ and $I_9 = E_{tent} \setminus \{(a, c)\} \cup \{e\}$, we have $az^{I_8} = az^{I_9}$ and then $bz^{I_8} = bz^{I_9}$. This implies that $b(e) = 0$.

We set $b(e) = \rho$ for $e \in E_{tent} \setminus E_{tent}^h$ and the proof is complete. □

Proof of Proposition 12 Remark that if $m = 2$, then a solution can be given only by a forest in the subgraph $G(K)$. We deduce that the maximum number of edges in this subgraph is equal to $|V(K)| - 1$.

Let us denote by $az \leq \alpha$ the inequality (9) associated with e . Let $bz \leq \beta$ be a facet defining inequality of $P_I(G, m)$ such that $\{z \in P_{\mathcal{I}}(G, m) : az = \alpha\} \subseteq \{z \in P_{\mathcal{I}}(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let $(u, v), (u, w) \in E(K)$, be two connected edges. Considering a line T_{uv} beginning by (u, v) and finishing by the vertex w and the line $T_{uw} = T_{uv} \setminus \{(u, v)\} \cup \{(u, w)\}$. These two solutions are feasible and satisfy the inequality (9) with equality. Moreover, we have $az^{T_{uv}} = az^{T_{uw}}$. Hence, $bz^{T_{uv}} = bz^{T_{uw}}$. This implies that $b((u, v)) = b((u, w))$. By symmetry, we deduce then that $b(e) = b(e')$ for all $e, e' \in E(K)$.

Let $e \in E \setminus E(K)$. By considering the feasible solution $T'_{uv} = T_{uv} \cup \{e\}$, we have $az^{T'_{uv}} = az^{T_{uv}}$. Hence, $bz^{T_{uv}} = bz^{T'_{uv}}$. This implies that $b(e) = 0$. \square

Proof of Proposition 13 Let us denote by $az \leq \alpha$ the inequality (10) associated with e . Let $bz \leq \beta$ be a facet defining inequality of $P_I(G, m)$ such that $\{z \in P_I(G, m) : az = \alpha\} \subseteq \{z \in P_I(G, m) : bz = \beta\}$. We show that $b = \rho a$ for some $\rho \in \mathbb{R}$.

Let e, e' be two edges in $E(K)$. The solution $I_1 = E(K) \setminus \{e\}$ and $I_2 = E(K) \setminus \{e'\}$ are feasible and verify the inequality (10) with equality. Hence, $az^{I_1} = az^{I_2}$. Therefore, $bz^{I_1} = bz^{I_2}$. This implies that $b(e) = b(e')$. We set $b(e) = \rho$. As $b(e)$ and $b(e')$ are arbitrary, then $b(e) = b(e')$ for all $e, e' \in E(K)$.

Let e_1 be an edge in $E \setminus E(K)$. The solution $I_3 = I_1 \cup \{e_1\}$ and I_1 are feasible and verify the inequality (10) with equality. Hence, $az^{I_1} = az^{I_3}$. Therefore $bz^{I_1} = bz^{I_3}$. This implies that $b(e_1) = 0$. As $b(e_1)$ is arbitrary we deduce that $b(e) = 0$, for all $e \in E \setminus E(K)$. Thus $b = \rho a$. \square

References

- Aardal K, Van Hoesel CPM (1999) Polyhedral techniques in combinatorial optimization II: applications and computations. *Stat Neerl* 53(2):131–177
- de CM Nogueira João Paulo, Arroyo José Elias C, Villadiego Harlem Mauricio M, Gonçalves Luciana B (2014) Hybrid GRASP heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties. *Electron Notes Theor Comput Sci* 302:53–72
- Gacias Bernat, Artigues Christian, Lopez Pierre (2010) Parallel machine scheduling with precedence constraints and setup times. *Comput Oper Res* 37(12):2141–2151
- Golumbic Martin Charles (2004) Algorithmic graph theory and perfect graphs, vol 57. Elsevier, New York
- Gonzales T, Sahni S (1976) Open shop scheduling to minimize finish time. *J ACM* 23:665–679
- Habib M, McConnell R, Paul C, Viennot L (2000) Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition, and consecutive ones testing. *Theor Comput Sci* 234:59–84
- Hassan Abdel-Jabbar M-A, Kacem I, Martin S (2016) *Lecture Notes of Computer. Science* 9849:308–319
- Hall Nicholas G, Posner Marc E (2001) Generating experimental data for computational testing with machine scheduling applications. *Oper Res* 49(7):854–865
- Jiinger M, Reinelt G, Thienel S (1995) Practical problem solving with cutting plane algorithms in combinatorial optimization. *Comb Optim Dimacs* 20:111–152
- Lekkekerker C, Boland J (1962) Representation of a finite graph by a set of intervals on the real line. *Fundam Math* 51(1):45–64
- Mokotoff E, Chretienne P (2002) A cutting plane algorithm for the unrelated parallel machine scheduling problem. *Eur J Oper Res* 141:515–525
- Pardalos PM, Xue J (1994) The maximum clique problem. *J Global Optim* 4(3):301–328
- Schrijver A (2003) *Combinatorial optimization : polyhedra and efficiency. Algorithms and combinatorics, vol 24.* Springer, Berlin
- Vallada E, Ruiz R (2012) Scheduling unrelated parallel machines with sequence dependent setup times and weighted earliness tardiness minimization. In *Just-in-Time systems*. Springer, New York, pp 67–90