Discrete Applied Mathematics 178 (2014) 46–57

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Characterizations of cographs as intersection graphs of paths on a grid

Elad Cohen [a,*], Martin Charles Golumbic [a], Bernard Ries [b,c]

[a] Caesarea Rothschild Institute and Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905, Israel

[b] PSL, Université Paris-Dauphine, 75775 Paris Cedex 16, France

[c] CNRS, LAMSADE UMR 7243, France

ARTICLE INFO

Article history:
Received 20 June 2013
Received in revised form 18 June 2014
Accepted 25 June 2014
Available online 26 July 2014

Keywords:
Cograph
Cotree
Grid
Intersection graph
Induced subgraph

ABSTRACT

A cograph is a graph which does not contain any induced path on four vertices. In this paper, we characterize those cographs that are intersection graphs of paths on a grid in the following two cases: (i) the paths on the grid all have at most one bend and the intersections concern edges ($\rightarrow B_1$-EPG); (ii) the paths on the grid are not bended and the intersections concern vertices ($\rightarrow B_0$-VPG).

In both cases, we give a characterization by a family of forbidden induced subgraphs. We further present linear-time algorithms to recognize B_1-EPG cographs and B_0-VPG cographs using their cotree.

1. Introduction

Edge intersection graphs of paths on a grid (or EPG graphs) are graphs whose vertices can be represented as paths on a rectangular grid such that two vertices are adjacent if and only if the corresponding paths share at least one edge of the grid. We may assume that the grid is $\mathbb{Z}^2$ or a sufficiently large subset of it. The EPG graphs were first introduced in [16] and have been studied by several authors (see for instance [2,3,5,22,23]). Every graph G is an EPG graph [16], so motivated by the study of these graphs with constraints from circuit layout problems, Golumbic, Lipshteyn and Stern introduced subclasses of EPG graphs based on restricting the number of bends permitted for each path. Specifically, for a fixed $k \geq 0$, the paths on the grid that represent the vertices of a graph are allowed to have at most k bends, i.e., at most k grid point turns, and the subclass of graphs that admit such a representation is denoted by B_k-EPG. Notice that B_0-EPG graphs are equivalent to interval graphs.

In [3], the authors show that for any k, only a small fraction of all labeled graphs on n vertices are B_k-EPG. Some results of [3] were also proved in [5]. In addition, the authors of [5] consider different classes of graphs and show, in particular, that every planar graph is a B_5-EPG graph. This result was later improved in [23], where the authors show that every planar graph is a B_4-EPG graph. It is still open if $k = 4$ is best possible. So far it is only known that there are planar graphs that are B_3-EPG graphs and not B_2-EPG graphs. The authors in [23] also show that all outerplanar graphs are B_2-EPG graphs thus proving a conjecture of [5].

For the case of B_1-EPG graphs, Golumbic, Lipshteyn and Stern [16] showed that every tree is a B_1-EPG graph, and Asinowski and Ries [2] showed that every B_1-EPG graph on n vertices contains either a clique or a stable set of size at least $n^{1/3}$. The problem of recognizing B_1-EPG graphs was shown to be NP-complete by Heldt, Knauer and Ueckerdt in [22]. It is

* Corresponding author. Tel.: +97 20545330023; fax: +972 48251045.
E-mail addresses: eladdc@gmail.com (E. Cohen), golumbic@cs.haifa.ac.il (M.C. Golumbic), bernard.ries@dauphine.fr (B. Ries).

http://dx.doi.org/10.1016/j.dam.2014.06.020
0166-218X/

therefore interesting to see which subfamilies of B_1-EPG graphs have special properties and which can be efficiently recognized. Asinowski and Ries [2] give a characterization of the B_1-EPG graphs among some subclasses of chordal graphs, namely, chordal bull-free graphs, chordal claw-free graphs, chordal diamond-free graphs, and special cases of split graphs. It follows from [16,3] that a complete bipartite graph $K_{m,n}$ ($m \le n$) is B_1-EPG if and only if $m \le 2$ and $n \le 4$. Since complete bipartite graphs are a special case of cographs, it is natural to ask for a characterization of B_1-EPG cographs. In [12], it is proven that cographs are well quasi ordered with respect to the induced subgraph relation. Therefore, any subfamily of the class of cographs can be characterized by a finite set of forbidden minimal induced subgraphs and recognized in polynomial time. However, it is only proven that such obstruction set exists. In Section 4 of this paper, we provide such a characterization for B_1-EPG by giving a complete family of minimal forbidden induced subgraphs. Later, in Section 6, we present an efficient linear-time algorithm to recognize this subfamily using their cotrees.

Instead of considering edge intersection graphs of paths on a grid, one may be interested in vertex intersection graphs of paths on a grid (or VPG graphs). The VPG graphs are graphs whose vertices correspond to paths on a rectangular grid such that two vertices are adjacent if and only if the corresponding paths share at least one grid point. These graphs were first introduced in [1] and have also been studied by several authors (see for instance [7,8,17]). In [1], the authors show that VPG graphs are exactly string graphs, i.e., intersection graphs of arbitrary curves in the plane. As in the case of EPG graphs, one may restrict the number of bends for each path. Hence, for a fixed $k \ge 0$, the paths on the grid that represent the vertices of a graph are allowed to have at most k bends, i.e., at most k grid point turns, and the subclass of graphs that admit such a representation is denoted by B_k-VPG. In [1], the authors notice that B_0-VPG graphs are equivalent to the so called 2-DIR graphs, whose recognition complexity is NP-complete [24].

A hierarchy of VPG graphs, relating them to other known families of graphs, is presented in [1], where they show for instance that planar graphs are B_3-VPG graphs. This result was recently improved in [8] where it was shown that planar graphs are B_2-VPG graphs. It remains open if $k = 2$ is best possible for planar graphs. In [17], the authors characterize split graphs that are B_0-VPG graphs by giving a family of minimal forbidden induced subgraphs. Furthermore, they characterize chordal claw-free B_0-VPG graphs as well as chordal bull-free B_0-VPG graphs. It is easy to see that all permutation graphs are B_1-VPG by labeling the x and y axes with the two permutations and connecting each pair of numbers with a single bend path. It thus follows that cographs (a subfamily of permutation graphs) are B_1-VPG. So it is natural to ask which cographs are B_0-VPG. In Section 5 of this paper, we characterize the B_0-VPG cographs as those which contain no induced 4-wheel, and present an efficient linear-time recognition algorithm using the cotree of the graph in Section 6.

We start with some preliminaries in Section 2, and in Section 3 we present some useful basic properties of the neighborhoods of C_4's in cographs which will be useful in our proofs characterizing B_1-EPG cographs. In Sections 4 and 5 we present characterizations for the classes of B_1-EPG cographs and B_0-VPG cographs, respectively. Linear time recognition algorithms for both of these classes are given in Section 6. Finally, we conclude with some open questions in Section 7. For graph theoretical terms that are not defined here, we refer the reader to [14,25].

2. Preliminaries

2.1. General graph definitions and notation

All graphs in this paper are connected, finite and simple. A *clique* is a set of pairwise adjacent vertices and a *stable set* is a set of pairwise nonadjacent vertices. The size of a maximum stable set in G is called the *stability number* of G and is denoted by $\alpha(G)$. A set $U \subseteq V$ is called *dominating* if for every vertex $v \in V \setminus U$ there exists $u \in U$ such that $uv \in E$. For disjoint sets $A, B \subseteq V$, we say that A is *complete* to B if every vertex in A is adjacent to every vertex in B, and that A is *anticomplete* to B if every vertex in A is nonadjacent to every vertex in B. The complement of a graph G will be denoted by $\overline{G}$. As usual, C_k, $k \ge 3$, denotes an induced cycle on k vertices. A vertex v which is adjacent to all the vertices of a C_k is called a *center*, and we call the graph induced by $V(C_k) \cup \{v\}$ a *k-wheel* denoting it by W_k (although it has $k + 1$ vertices). Finally, P_k, $k \ge 0$, denotes an *induced path* on k vertices, K_p, $p \ge 0$, denotes a *clique* on p vertices, mK_p, $m, p \ge 0$, denotes m disjoint copies of mK_p, and $K_{p,q}$ denotes the *complete bipartite graph* with p vertices in one set of the bipartition and q vertices in the other set of the bipartition. More generally, $K_{m_1,\dots,m_t}$ is the *complete multipartite graph* with part-sizes $m_1, \dots, m_t$.

Let $G = (V, E)$ be a graph. For a vertex $v \in V$, we let $\mathcal{N}_G(v)$ denote the set of vertices in G that are adjacent to v, i.e., the neighbors of v. $\mathcal{N}_G(v)$ is called the *neighborhood* of vertex v. We will write $\mathcal{N}_G[v] = N_G(v) \cup \{v\}$, and call $\mathcal{N}_G[v]$ the *closed neighborhood* of vertex v. Whenever it is clear from the context what G is, we will drop the subscripts and write $\mathcal{N}(v) = \mathcal{N}_G(v)$ and $\mathcal{N}[v] = \mathcal{N}_G[v]$. A vertex v is called a *true twin* of some vertex u if $\mathcal{N}[v] = \mathcal{N}[u]$. We will denote by $G[X]$ the subgraph induced by $X \subseteq V$. We write $G - v$ for the subgraph obtained by deleting vertex v and all the edges incident to v. Similarly, for $A \subseteq V$, we denote by $G - A$ the subgraph of G obtained by deleting the set A and all the edges incident to some vertex in A, i.e., $G - A = G[V \setminus A]$.

We will denote by G_R the *reduced graph* of G, that is, the graph obtained from G by deleting for each set U of true twins all but one $u \in U$. Thus, G_R does not contain any pair of adjacent vertices which have exactly the same closed neighborhood. The next lemma immediately follows from the definition of the reduced graph G_R.

Lemma 1. *Let G be a graph and let G_R be its reduced graph. Then any connected component of G_R isomorphic to a clique is an isolated vertex.*

2.2. EPG and VPG graphs

Let $\mathcal{G}$ be a rectangular grid. Let $\mathcal{P}$ be a collection of nontrivial simple paths on $\mathcal{G}$. As in [16], we define the *edge intersection graph EPG($\mathcal{P}$) of $\mathcal{P}$* to have vertices which correspond to the members of $\mathcal{P}$, so that two vertices are adjacent in *EPG*($\mathcal{P}$) if and only if the corresponding paths in $\mathcal{P}$ share at least one edge in $\mathcal{G}$. An undirected graph G is called an *edge intersection graph of paths on a grid (EPG)* if $G = EPG(\mathcal{P})$ for some $\mathcal{P}$ and $\mathcal{G}$, and $\langle \mathcal{P}, \mathcal{G} \rangle$ is an *EPG representation* of G.

Similar to EPG graphs, following [1], we define the *vertex intersection graph VPG($\mathcal{P}$) of $\mathcal{P}$* as the graph whose vertices correspond to the members of $\mathcal{P}$, and such that two vertices are adjacent in *VPG*($\mathcal{P}$) if and only if the corresponding paths in $\mathcal{P}$ share at least one grid point in $\mathcal{G}$. An undirected graph $G = (V, E)$ is called a *vertex intersection graph of paths on a grid (VPG)* if $G = VPG(\mathcal{P})$ for some $\mathcal{P}$ and $\mathcal{G}$, and $\langle \mathcal{P}, \mathcal{G} \rangle$ is a *VPG representation* of G.

For any vertex $v \in V$, we denote by P_v the corresponding path in the EPG (resp., VPG) representation of G. In this paper, we will always assume that the size of the grid $\mathcal{G}$, in particular m, is sufficiently large such that the EPG (resp., VPG) graphs that we are interested in admit an EPG (resp., VPG) representation in $\mathcal{G}$.

A turn of a path at a grid point is called a *bend* and the grid point is called a *bend point*. An EPG (resp., VPG) representation is *B_k-EPG* (resp., *B_k-VPG*) if each path has at most k bends. A graph that has a B_k-EPG (resp., B_k-VPG) representation is called *B_k-EPG* (resp., *B_k-VPG*). In this paper, we are interested in B_1-EPG graphs and B_0-VPG graphs which are also cographs.

Consider a rectangular grid of size $(2m+1) \times (2m+1)$. The horizontal grid lines will be referred to as *rows* and denoted by $y_{-m}, y_{-m+1}, \ldots, y_0, \ldots, y_{m-1}, y_m$, and the vertical grid lines will be referred to as *columns* and denoted by $x_{-m}, x_{-m+1}, \ldots, x_0, \ldots, x_{m-1}, x_m$. We define a $\ulcorner$ *-path* P as a single bended path with bend point (x_i, y_j) such that P uses column x_i between rows y_k and y_j, for some $j > k$, and P uses row y_j between columns x_i and x_l, for some $l > i$. In a similar way, $\urcorner$ *-paths*, $\llcorner$ *-paths*, and $\lrcorner$ *-paths* are defined. We say that a path P on the grid *contains* a grid point (x_i, y_j) if $(x_i, y_j) \subseteq P$ and (x_i, y_j) is not an endpoint of P.

The next observation shows that, when convenient, we do not need to consider true twins.

Lemma 2 (*True Twin Lemma*). *Let $G = (V, E)$ be a graph. Let $v \in V$ be a true twin of $u \in V$. Then G is B_k-EPG (resp., B_k-VPG) if and only if $G - v$ is B_k-EPG (resp., B_k-VPG).*

Proof. Consider a graph $G = (V, E)$ containing a true twin $v \in V$ of some vertex $u \in V$. Clearly, if G is B_k-EPG (resp., B_k-VPG), then $G - v$ is also B_k-EPG (resp., B_k-VPG). So assume now that $G - v$ is B_k-EPG (resp., B_k-VPG) and consider a B_k-EPG representation (resp., a B_k-VPG representation) of $G - v$. We then simply add a copy P_v of the path P_u representing vertex v. Since $\mathcal{N}[u] = \mathcal{N}[v]$, this gives us a feasible B_k-EPG representation (resp., a B_k-VPG representation) of G. □

3. Neighborhood properties of C_4's in cographs

Consider two graphs G and H. We say that G is *H-free*, if G does not contain any induced subgraph isomorphic to H. Let $\mathcal{F}$ be a family of graphs. G is said to be *$\mathcal{F}$-free* if G is H-free for every graph H of $\mathcal{F}$.

A graph G is a *cograph* if G is P_4-free. Cographs (*complement reducible* graphs) were originally defined recursively as follows:

(1) a single vertex is a cograph;
(2) the disjoint union of cographs is a cograph;
(3) the join of disjoint cographs is a cograph,

where the *join* of disjoint graphs $G_1, \ldots, G_k$ is the graph G with

$$V(G) = V(G_1) \cup \cdots \cup V(G_k) \quad \text{and}$$

$$E(G) = E(G_1) \cup \cdots \cup E(G_k) \cup \{xy \mid x \in V(G_i), y \in V(G_j), i \neq j\}.$$

The recursive construction of a cograph can be recorded in a data structure known as the *cotree* which fully encodes the cograph, and is often used for algorithms applied to cographs, such as those we will see in Section 6.

Remark 3. The equivalence of cographs as defined by (1)–(3) and P_4-free graphs was given independently by Gurvich [18–20] and by Corneil, Lerchs and Burlingham [10] where further results on the theory of cographs were developed. See also [15].

We start by giving some preliminary results which will be used later in the main proofs. Let $G = (V, E)$ be a graph containing an induced 4-cycle C with vertex set $\{v_1, v_2, v_3, v_4\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_4v_1\}$. Let V_{123} be the set of vertices in $G - V(C)$ adjacent to each of v_1, v_2, v_3 and nonadjacent to v_4. We define V_{234}, V_{341} and V_{412} similarly. Let V_{13} be the set of vertices in $G - V(C)$ adjacent to both v_1, v_3 and nonadjacent to both v_2, v_4. We define V_{24} similarly. Finally, let V_{1234} be the set of centers of C, and let $\mathcal{N}(C)$ be the set of vertices having at least one neighbor in C. Note that, by definition, $V(C) \subseteq \mathcal{N}(C)$.

When G is a cograph, properties (a)–(f) below immediately follow from the fact that G has no induced P_4.

Lemma 4. *If G is a cograph containing an induced* 4-cycle *C with vertex set $\{v_1, v_2, v_3, v_4\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_4v_1\}$, then the following properties hold:*

Property (a) *A vertex v not in $V(C)$ cannot be adjacent to only one vertex of C or to only one edge of C.*
Property (b) *$V_{123}, V_{234}, V_{341}$ and V_{412} are complete to V_{1234}.*

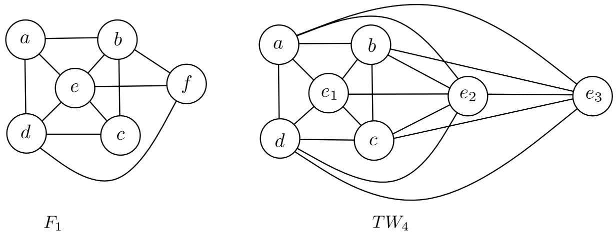


Fig. 1. The graphs $K_{3,2,1}$ and $K_{2,2,2,1}$.

Property (c) *$V_{123} \cup V_{341}$ is complete to $V_{234} \cup V_{412}$.*
Property (d) *V_{123} is anticomplete to V_{341}, and V_{412} is anticomplete to V_{234}.*
Property (e) *$\mathcal{N}(C) = V_{1234} \cup V_{13} \cup V_{24} \cup V_{123} \cup V_{234} \cup V_{341} \cup V_{412} \cup V(C)$.*
Property (f) *If $v \in V \setminus \mathcal{N}(C)$ then v is anticomplete to every set in $\mathcal{N}(C)$ except possibly to the set of centers V_{1234}.*
Property (g) *V_{13} is complete to V_{24}.*

The following lemma and its corollary will be used several times in our main proofs.

Lemma 5. *Let $G = (V, E)$ be a cograph containing an induced C_4 with vertex set $\{v_1, v_2, v_3, v_4\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_4v_1\}$. If G is either* (i) *W_4-free or* (ii) *$K_{3,2,1}$-free and $V_{1234} \neq \varnothing$, then all the vertices in V_{ijk} are true twins of v_j, for $ijk \in \{123, 234, 341, 412\}$.*

Proof. Let $u \in V_{ijk}$ and suppose that u is not a true twin of v_j. Then there exists a vertex w which is adjacent to u and nonadjacent to v_j (the case when w is adjacent to v_j and nonadjacent to u is symmetric). We claim that w must be adjacent to v_i and to v_k. Indeed, if w is nonadjacent to v_i (resp., v_k), then w must be adjacent to v_ℓ, $\ell \in \{1, 2, 3, 4\} \setminus \{i, j, k\}$, since otherwise $G[\{w, u, v_i, v_\ell\}]$ (resp., $G[\{w, u, v_k, v_\ell\}]$) is isomorphic to P_4, a contradiction. But now $G[\{w, v_\ell, v_i, v_j\}]$ (resp., $G[\{w, v_\ell, v_k, v_j\}]$) is isomorphic to P_4, a contradiction. Thus w must be adjacent to v_i, v_k as claimed. Hence, either $w \in V_{ik}$ or $w \in V_{k\ell i}$. It follows from Property (d) that w is nonadjacent to v_ℓ since $u \in V_{ijk}$. We conclude that $w \in V_{ik}$.

Case (i) If G is W_4-free, then we get a contradiction because $G[\{v_i, v_j, v_k, w, u\}]$ is isomorphic to W_4.

Case (ii) If G is $K_{3,2,1}$-free and $V_{1234} \neq \varnothing$, let $v \in V_{1234}$. Then it follows from Property (b) that u is adjacent to v. Also, w must be adjacent to v, otherwise $G[\{w, u, v, v_\ell\}]$ is isomorphic to P_4, a contradiction. But now $G[\{v_i, v_j, v_k, v_\ell, w, v\}]$ is isomorphic to $K_{3,2,1}$, a contradiction. Thus, in both cases, u must be a true twin of v_j. □

Corollary 6. *Let G be a connected and reduced cograph (i.e., with no true twins) containing an induced* 4-cycle *C with vertex set $\{v_1, v_2, v_3, v_4\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_4v_1\}$, then the following properties hold:*

(i) *If G is W_4-free, then $V(G) = \mathcal{N}(C) = V_{13} \cup V_{24} \cup V(C)$.*
(ii) *If $V_{1234} \neq \varnothing$ and G is $K_{3,2,1}$-free, then $\mathcal{N}(C) = V_{1234} \cup V_{13} \cup V_{24} \cup V(C)$, and $V_{13} \cup V_{24}$ is anticomplete to V_{1234}.*
(iii) *If G is $K_{3,3}$-free, then at least one of V_{13} or V_{24} is empty.*

Proof. (i) By Lemma 5, $V_{123}, V_{234}, V_{341}, V_{412} = \varnothing$ since G has no true twins. If G is W_4-free, then $V_{1234} = \varnothing$, so by Property (e), $\mathcal{N}(C) = V_{13} \cup V_{24} \cup V(C)$. Since G is connected, it follows from Property (f) that $V(G) \setminus \mathcal{N}(C)$ is empty.

(ii) Next, we assume $V_{1234} \neq \varnothing$ and G is $K_{3,2,1}$-free. Property (e) and Lemma 5 prove the first claim. Suppose $u \in V_{1234}$ and $w \in V_{24}$. If u and w are adjacent, then $G[\{v_1, v_2, v_3, v_4, u, w\}]$ is isomorphic to $K_{3,2,1}$, a contradiction. So V_{24} is anticomplete to V_{1234}. Similarly for V_{13}.

(iii) Finally, assume that G is $K_{3,3}$-free. By Property (g), V_{13} is complete to V_{24}. If both are nonempty, then together with $\{v_1, v_2, v_3, v_4\}$ we would have an induced $K_{3,3}$, a contradiction. □

4. Characterizing B_1-EPG cographs

In this section, we will characterize cographs which are B_1-EPG. It was shown in [16] that $K_{3,3}$ is not B_1-EPG and in [3] that $K_{2,5}$ is not B_1-EPG. Consider now the following two complete multipartite graphs (see Fig. 1): $K_{3,2,1}$ and $K_{2,2,2,1}$.

Lemma 7. *The graphs $K_{3,2,1}$ and $K_{2,2,2,1}$ are not B_1-EPG.*

Proof. Let us try to construct a B_1-EPG representation. Consider the induced 4-cycle $C = \{ab, bc, cd, ad\}$. It was shown in [16] that C must be represented either as a true pie, a false pie, or a frame (see Fig. 2). Let e be a center of C. It is easy to see that C cannot be represented as a frame since otherwise P_e cannot intersect all four paths P_a, P_b, P_c, P_d. Thus, we may assume that C is represented as a true or false pie centered at grid point (x_i, y_j).

Consider the graph $K_{2,2,2,1}$. If C is represented as a true pie, then every center of C corresponds either to a path using column x_i and containing but not bending at (x_i, y_j) or to a path using row y_j and containing but not bending at (x_i, y_j). Thus, in this case, the centers e_1, e_2, e_3 cannot induce a P_3, a contradiction. If C is represented as a false pie, then we may assume without loss of generality that P_a is a $\lrcorner$-path and P_c is a $\ulcorner$-path and both have bend point (x_i, y_j). Then every center of C

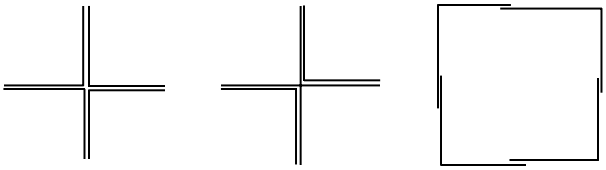


Fig. 2. True pie (left), false pie (middle), frame (right).

must correspond either to a $\llcorner$ -path with bend point (x_i, y_j) or to a $\urcorner$ -path with bend point (x_i, y_j). Clearly, in this case too, it is not possible for the centers e_1, e_2, e_3 to induce a P_3. Thus, $K_{2,2,2,1}$ is not B_1-EPG.

Let us now consider $K_{3,2,1}$. If C is represented as a true pie, then clearly we cannot add P_f since it must contain the bend point (x_i, y_j) in order to intersect P_b and P_d, and hence it would share a grid edge with at least one of P_a, P_c. If C is represented as a false pie, we may assume as before that P_a is a $\lrcorner$ -path, P_c is a $\ulcorner$ -path, both having bend point (x_i, y_j). Furthermore, P_b and P_d contain but do not bend at (x_i, y_j) and we may assume that P_b uses column x_i and P_d uses row y_j. Now consider the center e. P_e must be a $\llcorner$ -path or a $\urcorner$ -path with bend point (x_i, y_j). But now clearly we cannot add P_f since it must contain the grid point (x_i, y_j) in order to intersect all three paths P_b, P_d, P_e, and hence it would intersect at least one of P_a or P_c, a contradiction. Thus $K_{3,2,1}$ is not B_1-EPG. □

We now present our characterization theorem.

Theorem 8. *Let $G = (V, E)$ be a cograph. Then G is a B_1-EPG graph if and only if G is $\{K_{3,3}, K_{2,5}, K_{3,2,1}, K_{2,2,2,1}\}$-free.*

Proof. If G is B_1-EPG, then it follows from [3,16] and from Lemma 7 that G does not contain any of $K_{3,3}, K_{2,5}, K_{3,2,1}, K_{2,2,2,1}$ as an induced subgraph. Conversely, let $G = (V, E)$ be a cograph which is $\{K_{3,3}, K_{2,5}, K_{3,2,1}, K_{2,2,2,1}\}$-free. Without loss of generality, we may assume that G is connected, and by Lemma 2 that G is reduced (i.e., has no true twins).

If G is C_4-free, then G is an interval graph and the result holds. Thus, we may assume now that G does contain induced 4-cycles. Let C be such a cycle with vertex set $\{v_1, v_2, v_3, v_4\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_4v_1\}$. Furthermore, by Corollary 6(iii) since G is $K_{3,3}$-free, we may assume, without loss of generality, that $V_{13} = \emptyset$.

We will distinguish two cases.

Case 1: *No induced C_4 in G contains a center.*

Notice that in this case G is W_4-free. From Corollary 6(i) and our assumption above that $V_{13} = \emptyset$, we have $V(G) = V_{24} \cup V(C)$. Let $v_5, v_6 \in V_{24}$. If v_5, v_6 are adjacent, then consider the induced 4-cycle $\{v_1v_2, v_2v_5, v_5v_4, v_4v_1\}$. Since we may assume by Lemma 5 that $V_{254} = \emptyset$, it follows that we may delete v_6 (it would be a true twin of v_5). Thus V_{24} is an independent set. Since G does not contain any induced subgraph isomorphic to $K_{2,5}$, it follows that $|V_{24}| \leq 2$. We conclude from the above that G is isomorphic either to C_4, or to $K_{2,3}$ or to $K_{2,4}$. Since all these graphs are clearly B_1-EPG, it follows that G is B_1-EPG.

Case 2: *C contains a center, i.e., $V_{1234} \neq \emptyset$.*

From Corollary 6(ii) and our assumption above that $V_{13} = \emptyset$, we have $\mathcal{N}(C) = V_{1234} \cup V_{24} \cup V(C)$, and that V_{24} is anticomplete to V_{1234}. Now we claim the following.

Claim (i): *$G[V_{1234}]$ is the disjoint union of two cliques that are anticomplete to each other.* In order to prove the claim, we need to show that $G[V_{1234}]$ does not contain neither $3K_1$ nor P_3 (this follows from the fact that the disjoint union of two cliques is the complement of a complete bipartite graph). Since G is $K_{2,2,2,1}$-free, $G[V_{1234}]$ does not contain P_3. Thus, we only need to prove that $G[V_{1234}]$ does not contain three vertices e_1, e_2, e_3 which are pairwise nonadjacent. Suppose that three such vertices exist. But then $G[\{v_2, v_3, v_4, e_1, e_2, e_3\}]$ is isomorphic to $K_{3,2,1}$, a contradiction. *This proves Claim* (i).

We now denote by K_C^1 and K_C^2 the two cliques in $G[V_{1234}]$ (notice that one of these cliques may be empty). Let us distinguish two subcases.

Subcase 2a: $V_{24} \neq \emptyset$.

Claim (ii): *$V(C)$ is a dominating set (i.e., $V \setminus \mathcal{N}(C) = \emptyset$).* Indeed, suppose that $V(C)$ is not a dominating set. Let $w \in V_{24}$. Since G is connected, there exists a vertex $u \in V \setminus \mathcal{N}(C)$ adjacent to some vertex $v \in \mathcal{N}(C)$. It follows from the above and from Property (f) that we necessarily have $v \in V_{1234}$. But then $G[\{u, v, v_2, w\}]$ is isomorphic to P_4, a contradiction. Thus $V(C)$ is a dominating set. *This proves Claim* (ii).

We conclude from Claim (ii) that we may assume in Subcase 2a that $V = V_{1234} \cup V_{24} \cup V(C)$. This together with Claim (i) and the fact that V_{24} is anticomplete to V_{1234} implies that the vertices in K_C^i, for $i = 1, 2$, are all pairwise twins and hence we may assume that $|K_C^i| \leq 1$, for $i = 1, 2$. Next we will show the following.

Claim (iii): *$G[V_{24}]$ is isomorphic to either* (1) K_1, *or* (2) $2K_1$, *or* (3) P_3, *or* (4) C_4. First notice that $G[V_{24}]$ has stability number at most two. Indeed, if $u_1, u_2, u_3 \in V_{24}$ are three pairwise nonadjacent vertices, then $G[\{v_1, v_2, v_3, v_4, u_1, u_2, u_3\}]$ is isomorphic to $K_{2,5}$, a contradiction. Hence $G[V_{24}]$ has at most two connected components, and if it has exactly two connected components, then both must be a clique. It follows from Lemma 1 that in this case both cliques have size one, and thus outcome (2) holds. So we may assume now that $G[V_{24}]$ is connected. Notice that since $G[V_{24}]$ is an induced subgraph of a cograph, it is also a cograph.

If $|V_{24}| = 1$, then we clearly have outcome (1). We cannot have $|V_{24}| = 2$, since $G[V_{24}]$ is not a clique (by Lemma 1). If $|V_{24}| = 3$, then outcome (3) holds since P_3 is the only connected graph on three vertices which is not a clique. Similarly, it is not difficult to check that C_4 is the only connected graph on 4 vertices which is P_4-free with stability number at most 2 and not containing any true twins. So outcome (4) holds if $|V_{24}| = 4$. Suppose now that $|V_{24}| \geq 5$. Since $G[V_{24}]$ is not a clique, it follows that there exist at least two vertices $u_1, u_2 \in V_{24}$ which are non-adjacent. Therefore, $\alpha(G[V_{24}]) = 2$, and we can partition the vertices in $V_{24} \setminus \{u_1, u_2\}$ as follows: $V(u_i)$ is the set of vertices which are adjacent to u_i and nonadjacent to $\{u_1, u_2\} \setminus \{u_i\}$, for $i = 1, 2$; $V(u_1, u_2)$ is the set of vertices being adjacent to both u_1 and u_2. Since $\alpha(G[V_{24}]) = 2$, it follows that $V(u_i)$ induces a clique, for $i = 1, 2$. Since $G[V_{24}]$ is a cograph, it immediately follows that $V(u_i)$ is complete to $V(u_1, u_2)$ for $i = 1, 2$ and that $V(u_1)$ is anticomplete to $V(u_2)$. But now every vertex $v \in V(u_i)$ is a true twin of u_i, for $i = 1, 2$. Thus $V(u_1) = V(u_2) = \varnothing$. Finally, since $|V_{24}| \geq 5$ and since $\alpha(G[V_{24}]) = 2$ and $G[V_{24}]$ does not contain any true twins, it follows that there must exist vertices $w_1, w_2, w_3 \in V(u_1, u_2)$ which induce a P_3. But now $G[\{v_2, v_4, u_1, u_2, w_1, w_2, w_3\}]$ is isomorphic to $K_{2,2,2,1}$, a contradiction. *This proves Claim* (iii).

We have now shown in Claim (iii), that we may assume in Subcase 2a that $G[V_{24}]$ is isomorphic either to K_1, or to $2K_1$, or to P_3, or to C_4. This allows us to construct a B_1-EPG representation of G as shown in Fig. 3. The paths $P_{V_{24}}$ allow to represent either K_1 or $2K_1$ or P_3 or C_4. Hence, G is B_1-EPG.

Subcase 2b: $V_{24} = \varnothing$.

Suppose that K_C^1 and K_C^2 are both non-empty. Let $v_5 \in K_C^1$ and $v_6 \in K_C^2$. Consider the 4-cycle C' induced by v_2, v_5, v_4, v_6. The vertices v_1 and v_3 are centers of C'. If $V_{56} \neq \varnothing$, then we are exactly in Subcase 2a with C' playing the role of C. So we may assume now that $V_{56} = \varnothing$. Recalling that $\mathcal{N}(C) = V_{1234} \cup V(C)$, we notice that $V_{254} = V_{462} = \varnothing$ (see Lemma 5). Thus G consists of a C_4 with two nonadjacent centers and hence G is B_1-EPG (see Fig. 4).

From the arguments above, we may now assume, without loss of generality, that $K_C^2 = \varnothing$. We prove the following.

Claim (iv): *Let $w_1, w_2 \in K_C^1$. Then either $\mathcal{N}(w_1) \subset \mathcal{N}(w_2)$ or $\mathcal{N}(w_2) \subset \mathcal{N}(w_1)$.* Since w_1, w_2 are not true twins, without loss of generality, we may assume there exists a vertex u such that $w_1u \notin E$ and $w_2u \in E$. If $\mathcal{N}(w_1) \not\subset \mathcal{N}(w_2)$, then there exists a vertex v such that $w_2v \notin E$ and $w_1v \in E$. Note that u, v are not in $\mathcal{N}(C)$. If u, v are nonadjacent, then $G[\{u, w_2, w_1, v\}]$ is isomorphic to P_4, a contradiction. If u, v are adjacent, then $G[\{u, v, w_1, v_1\}]$ is isomorphic to P_4, a contradiction. Hence such a vertex v does not exist and so $\mathcal{N}(w_1) \subset \mathcal{N}(w_2)$. *This proves Claim* (iv).

It follows from Claim (iv) that there exists a vertex $w \in K_C^1$ such that $\mathcal{N}(u) \subseteq \mathcal{N}(w)$ for all $u \in K_C^1$. We claim that w is a universal vertex in G. Assume by contradiction that there exists a vertex v which is nonadjacent to w. Since G is connected we may assume that there exists z such that v, z, w induce a P_3. Note that v, z are not in $\mathcal{N}(C)$. Since $\mathcal{N}(u) \subseteq \mathcal{N}(w)$ for all $u \in K_C^1$, it follows that $z \notin K_C^1$. But now $G[\{v, z, w, v_1\}]$ is isomorphic to P_4, a contradiction.

Clearly, if $V \setminus \mathcal{N}(C)$ is empty, then G is B_1-EPG. So we may assume now that $V \setminus \mathcal{N}(C)$ is nonempty. Now suppose by contradiction that G is not B_1-EPG and let H be a minimal counterexample. Consider a B_1-EPG representation of $H \setminus \{v_1, v_2, v_3\}$ (which exists because of the minimality of H). It was shown in [16] that a clique K may be represented only in two ways: as an *edge-clique*, i.e., all paths representing vertices of K share at least one edge in the grid, or as a *claw-clique*, i.e., all paths representing vertices of K contain exactly two of the three edges of a given claw (a $K_{1,3}$) in the grid. Consider the clique $K_C^1 \cup \{v_4\}$. We may assume that the clique $K_C^1 \cup \{v_4\}$ is represented as an edge clique since w is a universal vertex. Indeed, suppose it is represented as a claw-clique with center (x_i, y_j). Since the path P_w must intersect all paths of the representation, it follows that either column x_i or row y_j at the left of x_i or row y_j at the right of x_i does not contain any paths representing vertices not belonging to the clique $K_C^1 \cup \{v_4\}$. Hence, we may delete that the corresponding part and transform the claw-clique into an edge-clique (see Fig. 5 for an example). Furthermore, since $K_C^1 \cup \{v_4\}$ is maximal, it follows that there exists at least one grid edge, say row y_i between columns x_j and x_{j+1}, such that only paths representing vertices in $K_C^1 \cup \{v_4\}$ use this edge. Clearly, by shifting and extending some paths if necessary, we may obtain a representation in which the paths representing vertices in $K_C^1 \cup \{v_4\}$ are the only paths using row y_i between columns x_j and x_{j+2}. But now we may delete P_{v_4} and add new paths $P_{v_1}, P_{v_2}, P_{v_3}, P_{v_4}$ to get a B_1-EPG representation of H by representing C as a true pie centered at grid point (x_{j+1}, y_i), a contradiction. This concludes the proof of Theorem 8. $\square$

5. Characterizing B_0-VPG cographs

In this section, we will give a characterization of cographs which are B_0-VPG.[1] It turns out that these graphs are much simpler to characterize and the only forbidden induced subgraph is the 4-wheel.

In [1], the authors proved the following.

Lemma 9 (*[1]*). *The graph C_4 has a unique B_0-VPG representation; it consists of two horizontal parallel paths intersecting with two vertical parallel paths.*

1 Note that all remaining cographs are B_1-VPG since the cographs are a subclass of permutation graphs which are all B_1-VPG.

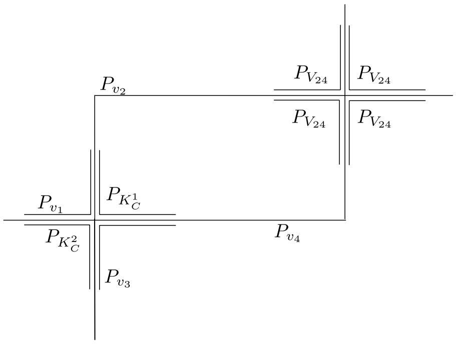


Fig. 3. B_1-EPG representation of G when $V_{24} \neq \emptyset$ and C contains a center.

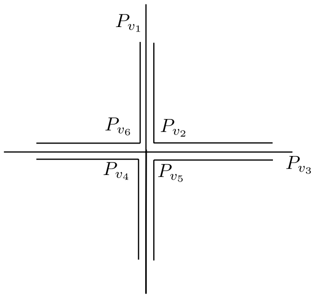


Fig. 4. B_1-EPG representation of G when $V_{24} = \emptyset$ and $K_C^1, K_C^2 \neq \emptyset$.

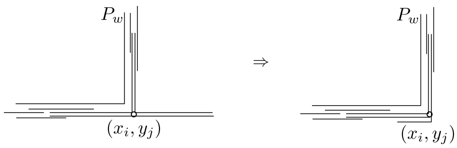


Fig. 5. Illustration of how to transform the claw-clique into an edge-clique for $K_C^1 \cup \{v_4\}$.

From this lemma, one can immediately conclude that the 4-wheel is not a B_0-VPG graph. But the following more general result holds.

Lemma 10. *W_k is not B_0-VPG for $k \geq 4$.*

Proof. It is easy to see that in any B_0-VPG representation of an induced cycle of length at least four, there must be two non-intersecting horizontal parallel paths as well as two non-intersecting vertical parallel paths. Consider W_k, $k \geq 4$, and let C_k be the induced k-cycle of the k-wheel. Since $k \geq 4$, it follows from the above that there exist at least four vertices, say v_1, v_2, v_3, v_4 (not necessarily consecutive on the cycle), such that the paths P_{v_1}, P_{v_3} lie on two distinct horizontal grid lines and the paths P_{v_2}, P_{v_4} lie on two distinct vertical grid lines. But then the path P_u corresponding to the center of C_k cannot intersect these four paths since it has no bends. Thus, W_k is not B_0-VPG. □

We can now state the characterization result for B_0-VPG cographs.

Theorem 11. *Let $G = (V, E)$ be a cograph. Then the following statements are equivalent:*

(i) *G is B_0-VPG;*
(ii) *G is W_4-free;*
(iii) *either G is an interval graph or G_R is a complete bipartite graph.*

Proof. Let G be a cograph. (i) ⇒ (ii) If G is B_0-VPG, then it follows from Lemma 10 that G is W_4-free.

(ii) ⇒ (iii) Suppose now that G is W_4-free. If G is C_4-free, then G is an interval graph. So we may assume now that G and hence G_R contains a C_4. Let C be an induced 4-cycle in G_R with vertex set $\{v_1, v_2, v_3, v_4\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_4v_1\}$. By Corollary 6 (i) we have that $V(G_R) = V_{13} \cup V_{24} \cup V(C)$, and by Property (g), V_{13} is complete to V_{24}. It is easy to see that V_{13} must be a stable set since two adjacent vertices $w_1, w_2 \in V_{13}$ would necessarily be true twins. By symmetry, the same holds for V_{24}. Thus, we obtain that G_R is a complete bipartite graph.

(iii) ⇒ (i) Since interval graphs and complete bipartite graphs are clearly B_0-VPG, and since the property of being B_0-VPG is maintained by adding true twins, the implication follows. □

6. Algorithmic aspects

In this section, we present efficient linear time algorithms for the recognition of B_0-VPG cographs and for B_1-EPG cographs. Since Theorems 8 and 11 give a characterization of B_1-EPG and B_0-VPG cographs, respectively, via a short list of forbidden induced subgraphs, it also immediately provides a polynomial-time algorithm to recognize such graphs. The brute force algorithm consisting of checking, for each forbidden subgraph, whether the input graph contains an induced subgraph isomorphic to it, trivially runs in time $O(n^7)$ in the case of B_1-EPG graphs and in time $O(n^5)$ in the case of B_0-VPG graphs. We will show that this can be improved by using the *cotree* of the input graph G.

In each case, we first run a linear time cograph recognition algorithm, such as those in [6,11,21], which also builds a cotree, a data structure that fully encodes the cograph. Once a cotree has been constructed for a cograph G, many familiar graph problems can be solved efficiently using a bottom-up calculation on the cotree, and in our particular case, the detection of the forbidden induced subgraphs characterizing B_0-VPG cographs and B_1-EPG cographs.[2]

6.1. The cotree representation of a cograph and recognizing B_0-VPG cographs

The recursive construction of a cograph G in a *bottom-up* fashion, according to rules (1)–(3) in Section 2, can be represented by a rooted tree T which records each union and join operation. The leaves of T are labeled by the vertices of G, and the subtree T_u rooted at an internal node u represents the subgraph G_u of G induced by the labels of its leaves, where u has as its children the roots of the subtrees of those disjoint cographs $H_1, \ldots, H_k$ $(k > 1)$ which were combined to form G_u. Moreover, u is labeled 0 if the *union* rule (2) was used, and labeled 1 if the *join* rule (3) was used. We call u a '0'-node or '1'-node according to its label.

Among all such constructions, there is a canonical one whose tree is called the *cotree* and satisfies the additional property that on every root-to-leaf path, the labels of the internal nodes alternate between 0 and 1, and every internal node has at least two children. The cotree can easily be obtained from any such 0/1 labeled tree T by coalescing all pairs of child–parent nodes in T having the same label, or where the parent has only one child. Vertices x and y of G are adjacent in G if and only if their least common ancestor in the cotree is labeled by 1. See [10].

We illustrate first how to find an induced W_4 of a cograph in linear time, if one exists, using a bottom-up calculation on the cotree. According to Theorem 11, this will recognize whether G is B_0-VPG.

An induced W_4 can appear in only a limited number of ways, namely, *join* $(2K_1, 2K_1, K_1)$, *join* $(K_{2,2}, K_1)$, or *join* $(K_{2,1}, 2K_1)$. This justifies the following algorithm, where red, orange, yellow, green and gold tokens stand for the increasing chain of induced subgraphs K_1, $2K_1$, $K_{2,1}$, $K_{2,2}$, $K_{2,2,1}$, (resp.) where the last graph $K_{2,2,1}$ is W_4.

Algorithm to find an induced W_4 in a cograph

Input: A cotree T for the cograph G.

Output: Confirmation that G is W_4-free, or an internal node of T marked with a gold token indicating an induced W_4 has been found.

Method: Mark each leaf with a red token [representing K_1].

Continuing bottom up on T: Find an internal node v whose children are marked,

If v is a '1'-node mark it as follows:

- **red** if all its children are red [contains a K_1]
- **yellow** if exactly one child is orange or yellow and all others are red [contains a $K_{2,1}$]
- **green** if exactly two children are orange and there are no other children [contains a $K_{2,2}$]
- **gold** if two children are orange and there is a third child of any color [contains a $K_{2,2,1}$]
- **gold** if one child is green and (by default) there is another child
- **gold** if one child is yellow and another is orange or yellow

if a gold token is produced, **exit** "Found an induced W_4" and return v.

If v is a '0'-node mark it as follows:

- **green** if it has a green child; else
- **yellow** if it has a yellow child; else
- **orange** (by default it has at least two children) [contains a $2K_1$]

if the root of the cotree is marked and no gold token has been produced, **exit** "G is W_4-free"

This algorithm is easily seen to be linear in the size of the cotree. In practice, if there are many copies of W_4, it will likely exit very quickly, assuming it finds one low in the cotree. However, it could run on the full cotree if the only W_4 appears only

[2] It should be pointed out that the problem of testing whether a cograph H occurs as an induced subgraph of a (larger) cograph G, when G and H are both part of the input, is NP-complete [13]. If the graph H is an arbitrary fixed cograph, then the complexity is clearly polynomial in the size of H, i.e. $|G|^{|H|}$, but it is not known whether it is linear or even FPL (Fixed Parameter Linear), see Section 7.

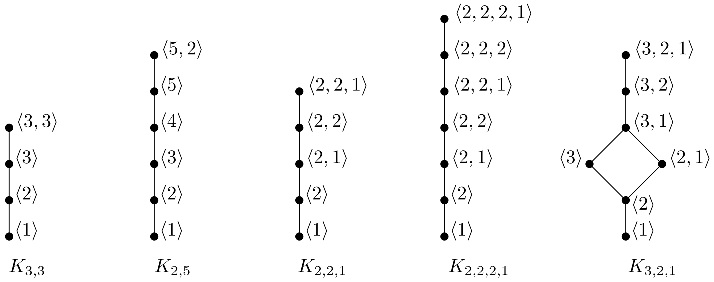


Fig. 6. The Hasse diagrams corresponding to the forbidden subgraphs $K_{3,3}$, $K_{2,5}$, $K_{2,2,1}$, $K_{2,2,2,1}$ and $K_{3,2,1}$, respectively.

at the root, or if the graph is W_4-free. In the following section we will generalize the algorithm and prove its correctness in order to recognize B_1-EPG graphs.

6.2. Recognizing B_1-EPG cographs

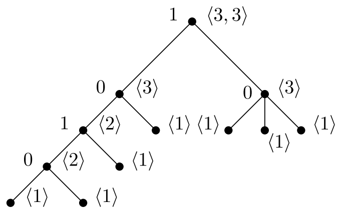
The recognition problem of B_1-EPG cographs is similar to the recognition of B_0-VPG cographs as in both cases the obstruction set consists of a small number of different complete multipartite graphs. Hence, the recognition problem reduces to the problem of detecting a complete multipartite graph in a cograph. Since a complete multipartite graph is a cograph it can be constructed using the join operation and the union operation, as follows: the union of k complete multipartite graphs $H_1, \ldots, H_k$ results in a graph consisting of the vertices and edges of $H_1, \ldots, H_k$, that is, no new complete multipartite graph is created except for a possibly larger independent set, while the join of $H_1, \ldots, H_k$ results in a new complete multipartite graph, whose maximal independent sets are precisely the maximal independent sets of each of $H_1, \ldots, H_k$.

We already saw in the previous section that a W_4, which can be also written as $K_{2,2,1}$, can be constructed in three different ways, namely, *join* $(2K_1, 2K_1, K_1)$, *join* $(K_{2,2}, K_1)$, or *join* $(K_{2,1}, 2K_1)$, in turn $K_{2,1}$, $K_{2,2}$ and $2K_1$ can be created exactly through *join* $(2K_1, K_1)$, *join* $(2K_1, 2K_1)$ and *union* (K_1, K_1), respectively. These configurations are the rules used to decide which token is chosen for each node of the cotree in the algorithm above for finding an induced W_4 in a cograph.

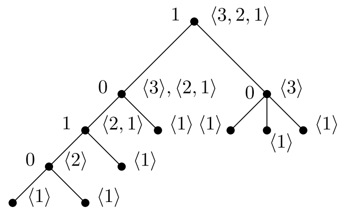
We denote a complete multipartite graph H by its non-increasing sorted vector $\langle m_1, m_2, \ldots, m_k\rangle$, where $\mathcal{M}(H) = \{m_i | i = 1, \ldots, k\}$ is the multiset of the sizes of the maximal independent sets of H (for instance the graph $K_{3,2,2,1}$ is denoted by $\langle 3, 2, 2, 1\rangle$ and $\mathcal{M}(H) = \{3, 2, 2, 1\}$). The relevant pieces for the construction of a complete multipartite graph H, namely, the *building blocks* of H, which we will denote by $\mathcal{BB}_H$, consists of (i) all induced subgraphs of H corresponding to complete multipartite graphs $K_{m_{i_1},\ldots,m_{i_\ell}}$ such that $\{m_{i_1}, \ldots, m_{i_\ell}\} \subseteq \mathcal{M}(H)$ and (ii) all independent sets of size at most the stability number of H, that is, $\{\langle k\rangle \mid k \leq \max(\mathcal{M}(H))\}$. Clearly, any building block can be built from its own set of building blocks. However, the set of building blocks is not closed under union or join. Indeed, for any complete multipartite graph H, there exists a complete multipartite graph H', which is built from the set $\mathcal{BB}_H$ and contains H as an induced subgraph, e.g., $H = K_{3,2,1}$ is an induced subgraph of $H' = K_{3,3,3}$, where $H' = join(3K_1, 3K_1, 3K_1)$ and $3K_1 \in \mathcal{BB}_H$. This motivates the following definition: A building block B' *dominates* a building block B, if there exists an injection $f : \mathcal{M}(B) \to \mathcal{M}(B')$, such that for every $x \in \mathcal{M}(B)$, $f(x) \geq x$. Note that B' dominating B also implies that B is an induced subgraph of B'. For example, the graph $K_{3,2}$ dominates the graph $K_{2,1}$, but the latter does not dominate the graph $3K_1$. For any complete multipartite graph H, the domination relation on the set $\mathcal{BB}_H$ defines a partially ordered set. In Fig. 6, we illustrate the Hasse diagram of the sets of building blocks of each of the forbidden subgraphs in Theorems 8 and 11.

Our algorithm is declarative, as in every node u of the cotree we search for its tokens which represent the constructed building blocks according to a set of rules. We summarize the set of rules for each of the forbidden subgraphs in Table 1 (the table of rules for a graph H is denoted by A_H), which specify the ways a building block can be constructed in a '1'-node and in a '0'-node by listing the configurations of the required minimal building blocks in the children of a node. That is, we search for the specified building blocks or building blocks which dominate them, e.g. if $H = K_{3,2,1}$ and the original graph G contains a $K_{2,2}$, we would like to mark it as containing the building block $K_{2,1}$. Each configuration is denoted by a sequence of building blocks concatenated with the $\bullet$ sign if it is a '1'-node or with the $\circ$ sign if it is a '0'-node. Instead of representing tokens with colors, we use the vector notation of a complete multipartite graph. We say that a token b' *dominates* another token b, if the building block corresponding to b' dominates the building block corresponding to b. For example, if we search for an induced $K_{2,2,1}$ and the current node v is a '1'-node and it has three children each marked with $\langle 2\rangle$, then v is marked with the token $\langle 2, 2, 1\rangle$ and $\langle 2\rangle \bullet \langle 2\rangle \bullet \langle 1\rangle$ is the corresponding configuration. As special cases, we denote $\langle 1\rangle \circ \langle 1\rangle \circ \cdots \circ \langle 1\rangle$, t times and $t \geq 2$, by $t \circ \langle 1\rangle$, rather than $\langle t\rangle$.

Note that the order of the rows in the table of rules of H corresponds to a topological sort of the Hasse diagram of H, in the sense that every descending induced chain of building blocks in the diagram corresponds to the increasing indices of the rows of the corresponding tokens. Moreover, in the cases of $K_{2,5}$, $K_{3,3}$, $K_{2,2,1}$ and $K_{2,2,2,1}$ the partially ordered sets are chains, while the case of $K_{3,2,1}$ is different, since it has width two, i.e., there are two building blocks which are incomparable with respect to inclusion, $K_{2,1}$ and $3K_1$. Hence, in the case of $K_{3,2,1}$, we need to have the space for two tokens in every internal node of the cotree. We are now able to describe the algorithm for detecting the forbidden subgraphs in Theorem 8, which



(a) Detecting a $K_{3,3}$.



(b) Detecting a $K_{3,2,1}$.

Fig. 7. A token marked cotree in the search cases of $K_{3,3}$ and $K_{3,2,1}$.

enables us to recognize B_1-EPG cographs. We assume that the given graph is a cograph and is associated with its cotree T. The algorithm consists of calling the following procedure where $A_H[i]$ denotes line i for table A_H. See Fig. 7 for two examples of a cotree marked with tokens according to the detection of a $K_{3,3}$ and a $K_{3,2,1}$.

Algorithm to find a $K_{2,5}$ or a $K_{3,3}$ or a $K_{2,2,1}$ or a $K_{2,2,2,1}$ or a $K_{3,2,1}$ in a cograph

Input: The cotree T for the cograph G, a table of rules A_H ($H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}, K_{3,2,1}\}$).

Output: Confirmation that G is H-free, or an internal node of T marked with a token representing H indicating an induced H has been found.

Method: Mark each leaf with a $\langle 1\rangle$ token; traverse bottom up on T and mark each internal node v (whose children have already been marked) with at most two tokens according to the table of rules.

Initialize $i := 0$.
Initialize *token* := *false*.
while no *token* was found **do**:
  if the children of v satisfy the rules in $A_H[i]$,
    then *token* := *true*,
      mark v with the token indicated in $A_H[i]$
  if $H == K_{3,2,1}$ and $A_H[i] == \langle 2, 1\rangle$ and $A_H[i+1] == \langle 3\rangle$,
    then mark v with a second token, indicated in $A_H[i+1]$
  $i := i + 1$
end
if a token representing H (i.e. $A_H[0]$) is produced,
  exit "Found an induced H" and return v
if v is the root of T and no token representing H has been produced,
  return "G is H-free",
else return

Theorem 12. *The algorithms above find an induced subgraph H in a cograph G if one exists, or declare that G is H-free, for $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}, K_{3,2,1}\}$.*

Proof. Let G be a cograph, T its cotree and $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}, K_{3,2,1}\}$. Suppose G contains an induced subgraph isomorphic to H. Then there is a subtree of T rooted in some internal node, whose leaves induce a subgraph of G that contains H. Let u be the lowest node in T, such that G_u contains H. Since H is connected, the node u is a '1' node, and since H is contained in G_u, there is a subset of the children of u, namely, $u_1, \ldots, u_k$, which represents induced subgraphs $H_1, \ldots, H_k$ of G, respectively, each dominating a building block of H, such that their join results in a graph H' which dominates H. Note that $k > 1$, otherwise u has a child u', such that $G_{u'}$ has as an induced subgraph isomorphic to H, contradicting the fact that u is the lowest such node in T.

We distinguish between the case of $H = K_{3,2,1}$ and the other cases, since the set of building blocks of H, for $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}\}$, forms a chain, while the set of building blocks of $K_{3,2,1}$ forms a partially ordered set of width 2.

In every iteration of the while loop we search for an induced subgraph that dominates a building block of H. In every processed row $A_H[i]$, we check if the rules are satisfied, i.e., if the children of the current node are marked with tokens that dominate the tokens in at least one of the configurations in $A_H[i]$. Every loop runs at most the number of rows in A_H, since every non-empty induced subgraph of G has $\langle 1\rangle$ as an induced subgraph.

In the case of $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}\}$, the increasing order of the rows in A_H corresponds to the decreasing order of the total order of the building blocks of H, starting from H itself. Hence, once a building block B is detected we can stop the loop, since B dominates all the building blocks in the following rows in A_H.

Table 1
Tables of rules for detecting a $K_{3,3}$, $K_{2,5}$, $K_{2,2,1}$, $K_{2,2,2,1}$ and $K_{3,2,1}$, respectively.

$A_{(3,3)}$	Token	'1'-node	'0'-node
0	$\langle 3, 3\rangle$	$\langle 3, 3\rangle$ or $\langle 3\rangle \bullet \langle 3\rangle$	$\langle 3, 3\rangle$
1	$\langle 3\rangle$	$\langle 3\rangle$	$\langle 3\rangle$ or $\langle 1\rangle \circ \langle 2\rangle$ or $3 \circ \langle 1\rangle$
2	$\langle 2\rangle$	$\langle 2\rangle$	$\langle 2\rangle$ or $2 \circ \langle 1\rangle$
3	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 1\rangle$

$A_{(2,5)}$	Token	'1'-node	'0'-node
0	$\langle 2, 5\rangle$	$\langle 2, 5\rangle$ or $\langle 2\rangle \bullet \langle 5\rangle$	$\langle 2, 5\rangle$
1	$\langle 5\rangle$	$\langle 5\rangle$	$\langle 5\rangle$ or $\langle 1\rangle \circ \langle 4\rangle$ or $\langle 2\rangle \circ \langle 3\rangle$ or $2 \circ \langle 1\rangle \circ \langle 3\rangle$ or $3 \circ \langle 1\rangle \circ \langle 2\rangle$ or $5 \circ \langle 1\rangle$
2	$\langle 4\rangle$	$\langle 4\rangle$	$\langle 4\rangle$ or $\langle 1\rangle \circ \langle 3\rangle$ or $\langle 2\rangle \circ \langle 2\rangle$ or $2 \circ \langle 1\rangle \circ \langle 2\rangle$ or $4 \circ \langle 1\rangle$
3	$\langle 3\rangle$	$\langle 3\rangle$	$\langle 3\rangle$ or $\langle 1\rangle \circ \langle 2\rangle$ or $3 \circ \langle 1\rangle$
4	$\langle 2\rangle$	$\langle 2\rangle$	$\langle 2\rangle$ or $2 \circ \langle 1\rangle$
5	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 1\rangle$

$A_{(2,2,1)}$	Token	'1'-node	'0'-node
0	$\langle 2, 2, 1\rangle$	$\langle 2, 2, 1\rangle$ or $\langle 2\rangle \bullet \langle 2\rangle \bullet \langle 1\rangle$ or $\langle 2, 2\rangle \bullet \langle 1\rangle$ or $\langle 2, 1\rangle \bullet \langle 2\rangle$	$\langle 2, 2, 1\rangle$
1	$\langle 2, 2\rangle$	$\langle 2, 2\rangle$ or $\langle 2\rangle \bullet \langle 2\rangle$	$\langle 2, 2\rangle$
2	$\langle 2, 1\rangle$	$\langle 2, 1\rangle$ or $\langle 2\rangle \bullet \langle 1\rangle$	$\langle 2, 1\rangle$
3	$\langle 2\rangle$	$\langle 2\rangle$	$\langle 2\rangle$ or $2 \circ \langle 1\rangle$
4	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 1\rangle$

$A_{(2,2,2,1)}$	Token	'1'-node	'0'-node
0	$\langle 2, 2, 2, 1\rangle$	$\langle 2, 2, 2, 1\rangle$ or $\langle 2\rangle \bullet \langle 2\rangle \bullet \langle 2\rangle \bullet \langle 1\rangle$ or $\langle 2, 2, 2\rangle \bullet \langle 1\rangle$ or $\langle 2, 2, 1\rangle \bullet \langle 2\rangle$ or $\langle 2, 2\rangle \bullet \langle 2, 1\rangle$	$\langle 2, 2, 2, 1\rangle$
1	$\langle 2, 2, 2\rangle$	$\langle 2, 2, 2\rangle$ or $\langle 2\rangle \bullet \langle 2\rangle \bullet \langle 2\rangle$ or $\langle 2, 2\rangle \bullet \langle 2\rangle$	$\langle 2, 2, 2\rangle$
2	$\langle 2, 2, 1\rangle$	$\langle 2, 2, 1\rangle$ or $\langle 2\rangle \bullet \langle 2\rangle \bullet \langle 1\rangle$ or $\langle 2, 2\rangle \bullet \langle 1\rangle$ or $\langle 2, 1\rangle \bullet \langle 2\rangle$	$\langle 2, 2, 1\rangle$
3	$\langle 2, 2\rangle$	$\langle 2, 2\rangle$ or $\langle 2\rangle \bullet \langle 2\rangle$	$\langle 2, 2\rangle$
4	$\langle 2, 1\rangle$	$\langle 2, 1\rangle$ or $\langle 2\rangle \bullet \langle 1\rangle$	$\langle 2, 1\rangle$
5	$\langle 2\rangle$	$\langle 2\rangle$	$\langle 2\rangle$ or $2 \circ \langle 1\rangle$
6	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 1\rangle$

$A_{(3,2,1)}$	Token	'1'-node	'0'-node
0	$\langle 3, 2, 1\rangle$	$\langle 3, 2, 1\rangle$ or $\langle 3\rangle \bullet \langle 2\rangle \bullet \langle 1\rangle$ or $\langle 3, 2\rangle \bullet \langle 1\rangle$ or $\langle 3, 1\rangle \bullet \langle 2\rangle$ or $\langle 2, 1\rangle \bullet \langle 3\rangle$	$\langle 3, 2, 1\rangle$
1	$\langle 3, 2\rangle$	$\langle 3, 2\rangle$ or $\langle 3\rangle \bullet \langle 2\rangle$	$\langle 3, 2\rangle$
2	$\langle 3, 1\rangle$	$\langle 3, 1\rangle$ or $\langle 3\rangle \bullet \langle 1\rangle$	$\langle 3, 1\rangle$
3	$\langle 2, 1\rangle$	$\langle 2, 1\rangle$ or $\langle 2\rangle \bullet \langle 1\rangle$	$\langle 2, 1\rangle$
4	$\langle 3\rangle$	$\langle 3\rangle$	$\langle 3\rangle$ or $\langle 1\rangle \circ \langle 2\rangle$ or $3 \circ \langle 1\rangle$
5	$\langle 2\rangle$	$\langle 2\rangle$	$\langle 2\rangle$ or $2 \circ \langle 1\rangle$
6	$\langle 1\rangle$	$\langle 1\rangle$	$\langle 1\rangle$

In the case of $H = K_{3,2,1}$, there are exactly two building blocks which are incomparable, namely $K_{2,1}$ and $3K_1$. If for some node v in T_u, G_v has two induced subgraphs isomorphic to $K_{2,1}$ and to $3K_1$, but has no induced subgraph isomorphic to a building block that dominates them, then we need to mark v with the appropriate tokens, namely, $\langle 2, 1\rangle$ and $\langle 3\rangle$. In the table of rules for $K_{3,2,1}$, the row of $\langle 3\rangle$ follows the row of $\langle 2, 1\rangle$. Therefore, when searching for a $K_{3,2,1}$ in G and a $K_{2,1}$ is detected, we check if the following row in the table of rules is satisfied and mark the node accordingly.

Once a building block is found it cannot be destroyed, since no edges are added to or deleted from the current constructed graph. Hence, when the algorithm reaches node u the subset of children $u_1, \ldots, u_k$ of u, are marked with the tokens which correspond to $H_1, \ldots, H_k$, respectively. Since u is a '1'-node and the join of $H_1, \ldots, H_k$ creates H' which dominates H, there is a configuration in $A_H[0]$ which is satisfied.

What remains to prove is that the table of rules of H, $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}, K_{3,2,1}\}$, is complete and correct. It is easy to check that for every building block in $\mathcal{BB}_H$ there is a row in A_H and that the rows are ordered according to the partial order of $\mathcal{BB}_H$. A trivial configuration for a token in A_H is the token itself, in other words we propagate building blocks from a child u to a parent u' in T, since G_u is an induced subgraph of $G_{u'}$. When processing a '1'-node, the configurations of a building block $B = \langle m_1, m_2, \ldots, m_k\rangle$ correspond to all splits of $\langle m_1, m_2, \ldots, m_k\rangle$ to smaller subsets. When processing a '0'-node, we can only create a larger independent set. Therefore, the configurations of a building block $B = \langle m\rangle$, correspond to all different integer sums of the number m. $\square$

We now turn to analyze the time and space complexity of the algorithm. Let G be a cograph and $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}, K_{3,2,1}\}$. We traverse every leaf once and every internal node twice, first when we process it and second when we treat it as a child. In each processed internal node, the loop runs at most the number of rows in the table of rules, which is constant. In each iteration, we check at most two rows in A_H. Since H is fixed and small, each row in A_H has a small number of configurations to test for. Also, testing if a configuration is satisfied and testing for domination between two building blocks is proportional to the size of H. For every node we store at most two tokens each of constant size. The cograph recognition problem and the construction of the cotree of a cograph takes linear time and space [6,11,21]. Therefore, together with Theorems 8 and 11, we have the following corollary.

Corollary 13. *The recognition problem for B_0-VPG cographs and B_1-EPG cographs can be solved in linear time and space.*

7. Conclusions and open questions

We characterized whether a cograph is a B_0-VPG graph or a B_1-EPG graph in terms of forbidden induced subgraphs and presented an algorithm to recognize these classes in linear time and space. The algorithm for detecting an induced subgraph $H \in \{K_{2,5}, K_{3,3}, K_{2,2,1}, K_{2,2,2,1}, K_{3,2,1}\}$ in a cograph can be generalized for an arbitrary fixed complete multipartite graph. The proposed algorithm in this paper is written in a declarative way, while an imperative algorithm is usually more efficient. In [9], we introduce an imperative algorithm for the generalized problem, where we first construct a hierarchical data structure of the building blocks, and then construct (bottom-up) in each node of the cotree its set of building blocks from those of its children, instead of iterating over all possible building blocks.

In [13], it is shown that the induced subgraph isomorphism problem for cographs is NP-complete, where G and H are part of the input. It would be interesting to know if the stronger problem of detecting a complete multipartite graph in a cograph remains NP-complete or if one can find a polynomial algorithm to solve it. The algorithm from [9] mentioned above shows that this problem is FPL (Fixed Parameter Linear). The same question could be asked for other special cases of cographs H beyond complete multipartite graphs. Similarly, one might investigate when G is restricted to be in some subfamilies of cographs, like threshold graphs and trivially perfect graphs. Note that, like in the case of Damaschke [13], it is known that induced subgraph isomorphism with H part of the input remains NP-complete even on connected trivially perfect graphs (Corollary 2 of [4]).

A permutation graph is the intersection graph of straight lines ending in two parallel lines, while a circle graph is the intersection graph of chords in a circle. It was shown in [1] that the class of circle graphs is contained in the class of B_1-VPG graphs. Since the class of cographs is contained in the class of permutation graphs which in turn is contained in the class of circle graphs, it will be interesting to study permutation B_0-VPG graphs. Similarly, for B_1-EPG graphs, the characterization problems for chordal B_1-EPG graphs and permutation B_1-EPG graphs remain open.

References

[1] A. Asinowski, E. Cohen, M.C. Golumbic, V. Limouzy, M. Lipshteyn, M. Stern, Vertex intersection graphs of paths on a grid, J. Graph Algorithms Appl. 16 (2012) 129–150.
[2] A. Asinowski, B. Ries, Some properties of edge intersection graphs of single-bend paths on a grid, Discrete Math. 312 (2012) 427–440.
[3] A. Asinowski, A. Suk, Edge intersection graphs of systems of paths on a grid with a bounded number of bends, Discrete Appl. Math. 157 (2009) 3174–3180.
[4] R. Belmonte, P. Heggernes, P. van't Hof, Edge contractions in subclasses of chordal graphs, Discrete Appl. Math. 160 (2012) 999–1010.
[5] T. Biedl, M. Stern, Edge intersection graphs of *k*-bend paths in grids, Discrete Math. Theor. Comput. Sci. 12 (2010) 1–12.
[6] A. Bretscher, D.G. Corneil, M. Habib, C. Paul, A simple linear time LexBFS cograph recognition algorithm, SIAM J. Discrete Math. 22 (2008) 1277–1296.
[7] S. Chaplick, E. Cohen, J. Stacho, Recognizing some subclasses of vertex intersection graphs of 0-bend paths in a grid, in: Proc. WG-2011, in: Lecture Notes in Computer Science, vol. 6986, Springer-Verlag, 2011, pp. 319–330.
[8] S. Chaplick, T. Ueckerdt, Planar graphs as VPG-graphs, in: Proc. GD-2012, in: Lecture Notes in Computer Science, vol. 7704, Springer-Verlag, 2013, pp. 174–186.
[9] E. Cohen, Intersection graphs of paths in a grid (Ph.D. thesis), University of Haifa, Israel, 2012.
[10] D. Corneil, H. Lerchs, L. Burlingham, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 163–174.
[11] D.G. Corneil, Y. Perl, L.K. Stewart, A linear time recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 924–934.
[12] P. Damaschke, Induced subgraphs and well-quasi-ordering, J. Graph Theory 14 (4) (1990) 427–435.
[13] P. Damaschke, Induced subraph isomorphism for cographs is NP-complete, Lecture Notes in Comput. Sci. 484 (1991) 72–78.
[14] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, second ed., in: Annals of Discrete Mathematics, vol. 57, Academic Press, New York, 1980, Elsevier, Amsterdam, 2004.
[15] M.C. Golumbic, V. Gurvich, Read-once functions, in: Y. Crama, P.L. Hammer (Eds.), Boolean Functions: Theory, Algorithms and Applications, Cambridge University Press, 2011, pp. 448–486.
[16] M.C. Golumbic, M. Lipshteyn, M. Stern, Edge intersection graphs of single bend paths on a grid, Networks 54 (2009) 130–138.
[17] M.C. Golumbic, B. Ries, On the intersection graphs of orthogonal line segments in the plane: characterizations of some subclasses of chordal graphs, Graphs Combin. 29 (2013) 499–517.
[18] V.A. Gurvich, On repetition-free Boolean functions, Uspekhi Mat. Nauk 32 (1977) 183–184 (in Russian), also, On read-once Boolean functions, Russian Math. Surveys 32 (1977) 183–184.
[19] V.A. Gurvich, Applications of Boolean functions and networks in game theory, Section 5, Repetition-free Boolean functionsand normal forms of positional games (Ph.D. thesis), Moscow Inst. of Physics and Technology, Moscow, USSR, 1978 (in Russian).
[20] V.A. Gurvich, Some properties and applications of complete edge-chromatic graphs and hypergraphs, Sov. Math. Dokl. 30 (3) (1984) 803–807.
[21] M. Habib, C. Paul, A simple linear time algorithm for cograph recognition, Discrete Appl. Math. 145 (2005) 183–197.
[22] D. Heldt, K. Knauer, T. Ueckerdt, Edge-intersection graphs of grid paths: the bend-number, Discrete Appl. Math. 167 (2014) 144–162.
[23] D. Heldt, K. Knauer, T. Ueckerdt, On the bend-number of planar and outerplaner graphs, in: Proceedings of 10th Latin American Symposium on Theoretical Informatics, LATIN 2012, pp. 458–469.
[24] J. Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, Discrete Appl. Math. 52 (1994) 233–252.
[25] D.B. West, Introduction to Graph Theory, second ed., Prentice Hall, 2001.