Universidad Nacional Autónoma de México

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS
Y DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

Minimal obstructions to $(s, 1)$*-polarity in cographs*

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS

PRESENTA:

FERNANDO ESTEBAN CONTRERAS MENDOZA

DIRECTOR DE TESIS:

DR. CÉSAR HERNÁNDEZ CRUZ
CINVESTAV

Ciudad de México, Noviembre de 2018

Para Betto,
mi eterno protector.

Contents

Introduction

Polar graphs are a natural extension of some well-studied classes of graphs which include bipartite graphs and its complements, as well as split graphs (i.e., graphs whose vertex set can be partitioned into a clique and a stable set). A graph G is called *polar* if its vertex set admits a bipartition such that one of the parts induces a complete multipartite graph and the other part induces a disjoint union of cliques. More specifically, we say that G is (s, k)-polar if there exist a partition $\{A, B\}$ of $V(G)$ where A induces a join of at most s stable sets and B induces the disjoint union of at most k cliques; such a partition is called an (s, k)-polar partition of G.

Since recognizing polar graphs is known to be an NP-complete problem, a natural question that arises is to find subclasses of polar graphs which can be recognized in polynomial time and for which nice characterizations can be found. It turns out that one can derive such results for a very interesting class of graphs: the cographs.

Cographs are the graphs without induced paths on four vertices. Many characterizations are known for cographs, for example, they are the graphs such that the complement of any connected induced subgraph is disconnected, or equivalently, the graphs that can be constructed from single vertex graphs by means of complement and disjoint union operations.

In 1990 it was shown that, on the class of cographs, every hereditary property of graphs can be characterized by a finite family of forbidden induced subgraphs [38]. The knowledge of such families has an interesting application, the use of its members in certifying algorithms for the associated decision problem. For example, *having an (s, k)-polar partition* is an hereditary property, thus, if we know the complete list of forbidden induced subgraphs that characterize cographs with such a property, an algorithm could be designed to receive a cograph G, decide if it has a partition of the desired type, and return either the partition of G (a yes-certificate) or one of the forbidden induced subgraphs

(a no-certificate). Thus, finding the family of forbidden induced subgraphs characterizing a given hereditary property in the class of cographs comes not only as a natural problem but as an interesting problem too.

Ekim, Mahadev and de Werra characterized the polar cographs in terms of forbidden induced subgraphs, as well as the (s, k)-polar cographs when $\min\{s, k\} \leq 1$ [47, 48]. They also proposed the problem of finding the corresponding characterization for the $(2, 2)$-polar cographs; this problem was recently solved by Hell, Hernández-Cruz and Linhares-Sales [74]. Additionally, Bravo, Nogueira, Protti and Vianna exhibited in [14] the family of minimal forbidden induced subgraphs for the $(2, 1)$-polar cographs, from where, by taking complements, it is trivial to obtain analogous results for $(1, 2)$-polar cographs.

In the present work, we will focus on the characterization of $(s, 1)$-polar cographs. As the main result, we develop a recursive complete characterization of the minimal forbidden induced subgraphs for the $(s, 1)$-polar cographs, and additionally, we make a quick estimation on the growth rate of the families of minimal obstruction showing a subexponential lower bound and a exponential upper bound (with an extremely bad overestimation). We also show that cographs having an $(s, 1)$-polar partition for some arbitrary non-fixed integer s, $s \geq 2$, can be characterized by a family of four forbidden induced subgraphs.

In the first chapter of this thesis we establish the basic notions on Graph Theory, with the purpose of that the text be self-contained; this chapter contains a breviary of graph classes that will be mentioned along the text, as well as a brief introduction of Computational Complexity Theory. In second chapter we tackle the polar graphs, introducing them in a wide context that includes *matrix partitions*, a kind of coloring generalizations. In the third chapter we introduce and delve on the class of cographs; in this chapter is contained a survey of the main characterizations of cographs, a straightforward development of the so-called *cotrees* (a structural representation of cographs by means of trees), and a small sample of some problems that are difficult to solve in general graphs, but are easy in cographs. Finally, in chapter four we focus on the characterization of $(s, 1)$-polar cographs; this chapter starts with a survey of closely related results on polar cographs, and then follows with some technical lemmas needed to prove our main results. Chapter four finish with a straightforward asymptotic estimation of the number of cograph minimal $(s, 1)$-polar obstructions. At the end of the text are presented the conclusions obtained from this work, as well as some future lines of work.

Chapter 1

Introduction to Graph Theory

Along this chapter will be established the basic terminology of Graph Theory with the purpose of this thesis be self-contained. All graphs in this document are considered to be finite and simple (without loops or multiple edges). In general terms, we follow [8] for the definitions, although some of them could be different.

1.1 Basic concepts and notation

A ***graph*** is an ordered pair $G = (V, E)$, where V is a finite non-empty set whose elements are called, the ***vertices*** of G while E is a subset of unordered pairs of distinct elements of V whose elements are called the ***edges*** of G, that is to say,

$$E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}.$$

The ***order*** and the ***size*** of a given graph, are the number of vertices and edges of that graph, respectively; n and m commonly will denote the order and the size of a graph. Graphs are commonly represented by drawings, as is shown in Figure 1.1.

For avoiding possible confusions, when we work with more than one graph, we will use a more specific notation to represent the parameters and elements of the graphs; for example, if G and H are graphs, we will use $V(G), E(G)$ and $V(H), E(H)$ to denote the vertex set and the edge set of G and H, respectively.

An unordered pair of distinct vertices $\{u, v\}$ of a graph G will be simply represented as uv. We say that vertices u and v of a graph G are ***adjacent*** if $uv \in E(G)$. In such a case, uv will be called the uv-edge of G, we will say

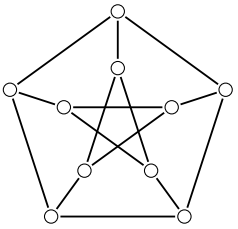


Figure 1.1: It is usual to represent the vertices of a graph with points (or little circles) and the edges with lines joining the corresponding points.

that u and v are the ***ends*** of uv and that uv is ***incident*** on its ends. Two vertices u and v of a graph G are called ***neighbors*** if $uv \in E$, and we will say that two edges that have a common end are ***adjacent***. Given a graph G, and a vertex v of G, the number of neighbors of v is the ***degree*** of v in G, and the ***neighborhood*** of v in G, denoted $N_G(v)$ or $N(v)$ when no confusion is possible, is the set of all its neighbors, that is,

$$N_G(v) = \{u \in V(G) \mid uv \in E(G)\}.$$

The ***maximum degree*** and the ***minimum degree*** on the vertices of a graph G are denoted by $\Delta(G)$ and $\delta(G)$, respectively. If $\Delta(G) = \delta(G) = k$ we will say that G is a k***-regular graph***, and if a graph G is k-regular for some integer k, G will be called a ***regular graph***.

For graphs G and H, a bijection $f : V(G) \to V(H)$ will be called an ***isomorphism*** if f is such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$; if a function f as above exists, we say that G and H are ***isomorphic graphs*** and we denote it by $G \cong H$. When we say that a given graph is unique with some specified property we are always referring to uniqueness up to isomorphism.

A graph H is said to be a ***subgraph*** of the graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. We denote the fact that H is a subgraph of G simply as $H \subseteq G$. If U is a non-empty subset of $V(G)$, the graph whose vertex set is U, and whose edge set is the set of all the edges of G such that both of its ends belongs to U, is called the ***induced subgraph*** by U in G, and it is denoted by $G[U]$. For a graph $G = (V, E)$ and a non-empty subset U of V, we use $G - U$ to denote the induced subgraph $G[V \setminus U]$; for a single vertex set $\{u\}$ we also use the notation $G - u$ instead of $G - \{u\}$. An example of an induced subgraph is shown in Figure 1.2.

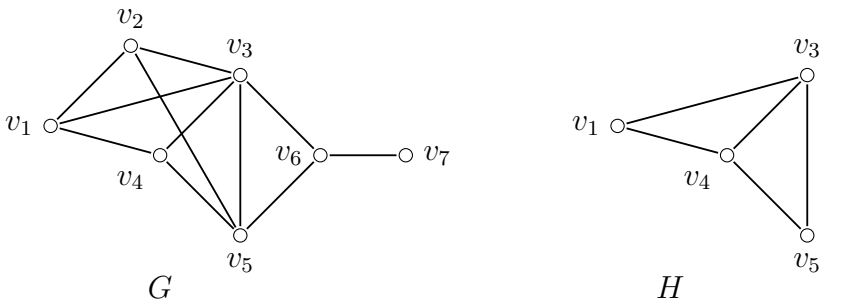


Figure 1.2: On the left is shown a graph G, on the right, the induced subgraph $H = G[\{v_1, v_3, v_4, v_5\}] = G - \{v_2, v_6, v_7\}$.

Given a graph H, we will say that the graph G is H***-free*** if G does not contain an isomorphic copy of H as induced subgraph; for a family of graphs $\mathcal{F}$, we say that G is $\mathcal{F}$***-free*** if it is F-free for every $F \in \mathcal{F}$. It is said that a property $\mathcal{P}$ of graphs is a ***hereditary property*** if, for every graph G that satisfies $\mathcal{P}$, any induced subgraph H of G also satisfies the property $\mathcal{P}$. Notice that if $\mathcal{P}$ is a hereditary property, then the graphs that satisfies $\mathcal{P}$ can be characterized by a (possibly infinite) family of forbidden induced subgraphs.

A ***walk*** W in a graph G is a finite sequence of vertices

$$W = (v_0, v_1, \ldots, v_{n-1}, v_n),$$

where v_i is neighbor of v_{i-1} for $i \in \{1, 2, \ldots, n\}$; the integer n is called the ***length*** of the walk W. If W is a walk that starts in the vertex u and ends in the vertex v, we will say that W is an uv***-walk***, or equivalently, that W is a walk *from* u *to* v. If each edge of W is traversed once, then W is called a ***trail*** and, if in addition $v_i \neq v_j$ for $i \neq j$, then we call it a ***path***; the path with n vertices (of length $n-1$) will be denoted by P_n. The same notions established for walks will be used for trails and paths, for example, an uv-path is a path that starts in u and whose end vertex is v.

It is said that a walk is ***closed*** if its initial vertex and its final vertex are the same. A ***cycle*** is a closed trail such that its vertices are distinct, except for the first and the last. For each integer n, $n \geq 3$, the only cycle of length n is denoted by C_n, and it is called the n-cycle. We say that a cycle is ***odd*** or ***even*** if its length is odd or even, respectively.

We say that vertices u and v of a graph G are *connected* if there is a uv-path in G, and we called the graph G ***connected*** if every pair of vertices of G are

connected. For a graph $G = (V, E)$, the relation on V given for every pair of vertices by the sentence "there exists an uv-walk in G" is an equivalence relation, and the graphs induced by each equivalence class are called the ***connected components*** of G, or simply, the ***components*** of G. Thus, a graph G is a connected graph if G has only one connected component; otherwise G is called ***disconnected***.

The ***distance*** $d_G(u, v)$ between two vertices, u and v, of a graph G is the minimum length on the uv-paths whenever u and v are connected, and it is infinite otherwise. The ***diameter*** of a graph G is the maximum distance between any pair of vertices of G. Equivalently, the diameter of a connected graph is the least positive integer d such that, between every pair of distinct vertices, u and v, there exists a uv-path in G whose length is at most d; the diameter of a disconnected graph is infinite.

A ***tree*** T is a graph such that any two vertices are connected by exactly one path, or equivalently, a connected graph without cycles as subgraphs. Given two vertices, u and v of a tree T, we denote the unique uv-path in T by uTv. A ***tree decomposition*** of a graph G is a tree T with vertex set $V(T) = (t_1, t_2, \ldots, t_n)$, such that t_i is a subset of $V(G)$ for any $i \in \{1, 2, \ldots, n\}$, and the three following properties are satisfied:

1. Each vertex of G is contained in at least one vertex of T, that is to say, $V(G) = \bigcup_{i=1}^{n} t_i$.

2. For every edge uv of G, there exists a vertex t_j of T such that $u, v \in t_j$.

3. For $i, j, k \in \{1, 2, \ldots, n\}$, if $t_j \in V(t_i T t_k)$, then $t_i \cap t_k \subseteq t_j$. That is to say, if two vertex of T, t_i and t_k, both contain a vertex v of G, then all vertices of T that belongs to the path $t_i T t_k$ contains v as well. Equivalently, for each vertex v of G, the graph induced by the vertices of T that contains v, is a connected subgraph of T.

Notice that conditions 1 and 2 together say that G is the union of the induced subgraphs $G[t_i]$; these subgraphs are usually called the parts of the tree decomposition. Condition 3 implies that the parts of the decomposition are organized roughly like a tree [41].

The ***width*** of a tree decomposition T is the number

$$\max\{|t| - 1 \mid t \in V(T)\},$$

and the ***treewidth*** of a graph G, $tw(G)$, is the least width of any tree decomposition of G. As one easily checks, trees themselves have treewidth one. An example of a graph and a tree decomposition of it with width two are shown in Figure 1.3.

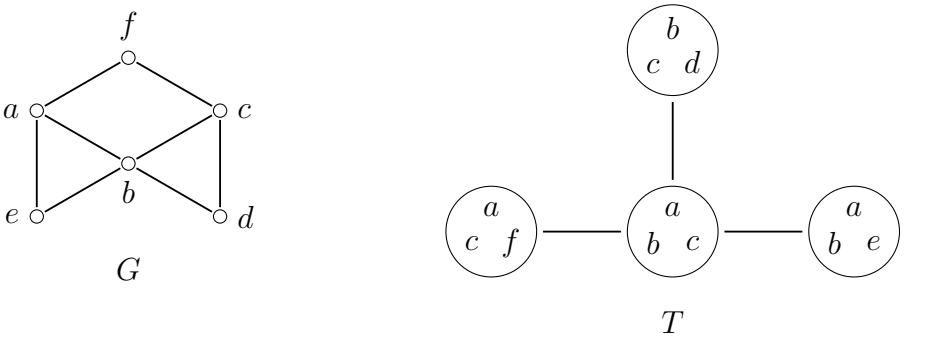


Figure 1.3: A graph G on six vertices and a tree decomposition T of G that shows that $tw(G) \leq 2$.

Given a graph $G = (V, E)$, a subset M of E is called a ***matching*** of G if any two elements of M are not adjacent. A matching M is a ***perfect matching*** if every vertex of G is incident in (exactly) one edge in M.

Sometimes, graphs receive special names for its apparent similarity with some objects of everyday life. Some of these graphs that will appear frequently in the text are depicted in Figure 1.4.

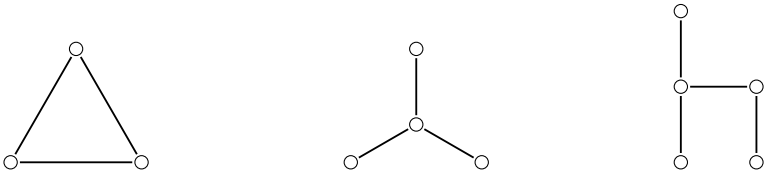


Figure 1.4: Because of its resemblance to certain objects, some graphs receive special names.

A ***directed graph***, or ***digraph*** in short, is an ordered pair $D = (V, A)$ where V is a finite non-empty set whose elements are called ***vertices***, and A

is a set of ordered pairs of vertices called ***arcs***. If $a = (u, v)$ is an arc of D then a is said to *join* u to v, the vertex u is the ***tail*** of a, and the vertex v its ***head***; u and v are the two ***ends*** of a. Similar terminology is used for graphs and digraphs, for example, if U is a non-empty subset of $V(D)$, the digraph whose vertex set is U, and whose arc set is the set of all the arcs of D such that both of its ends belongs to U, is called the ***induced subdigraph*** by U in D. A digraph is called ***transitive*** if, for distinct vertices u and w, (u, w) is an arc whenever (u, v) and (v, w) are arcs. One may obtain a digraph from a graph G by replacing each edge by just one of the two possible arcs with the same ends; such a digraph is called an ***orientation*** of G.

1.2 Graph operations

For each graph G it is defined the ***complement graph*** of G, denoted by $\overline{G}$, as the graph whose vertex set is $V(\overline{G}) = V(G)$ and whose edge set is given by $E(\overline{G}) = \{uv \mid uv \notin E(G)\}$.

Let G and H graphs such that $V(G)$ and $V(H)$ are disjoint sets. The ***disjoint union*** of G and H is the graph

$$G + H = (V(G) \cup V(H), E(G) \cup E(H)).$$

That is to say, the disjoint union of the graphs G and H is the graph whose vertex set is the union of the vertex sets of G and H, and whose edge set is the union of the edge sets of both graphs, with adding no more edges. For every graph G, the disjoint union of k vertex-disjoint copies of G, $G_1 + G_2 + \cdots + G_k$, will be denoted by kG.

The ***join*** of the vertex-disjoint graphs G and H is the graph $G \oplus H$ obtained by $G + H$ by adding all the possible edges with one end in $V(G)$ and the other end in $V(H)$, that is to say,

$$V(G \oplus H) = V(G) \cup V(H)$$

and

$$E(G \oplus H) = E(G) \cup E(H) \cup \{uv \mid u \in V(G), v \in V(H)\}.$$

Let G be a graph and let $u, v \in V(G)$. We say that u and v are ***twins*** if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$, that is, if they have the same neighbors, except possibly for themselves. Two twins are called ***true twins*** if they are adjacent

and ***false twins*** otherwise. We refer to the operation of adding one (true or false) twin to a vertex of a graph as a ***vertex doubling***.

If $G = (V, E)$ is a graph, the ***line graph*** of G is the graph $L(G)$, with vertex set E and such that two of its vertices are adjacent if and only if the corresponding edges of G are adjacent. A graph and its corresponding line graph are shown in Figure 1.5.

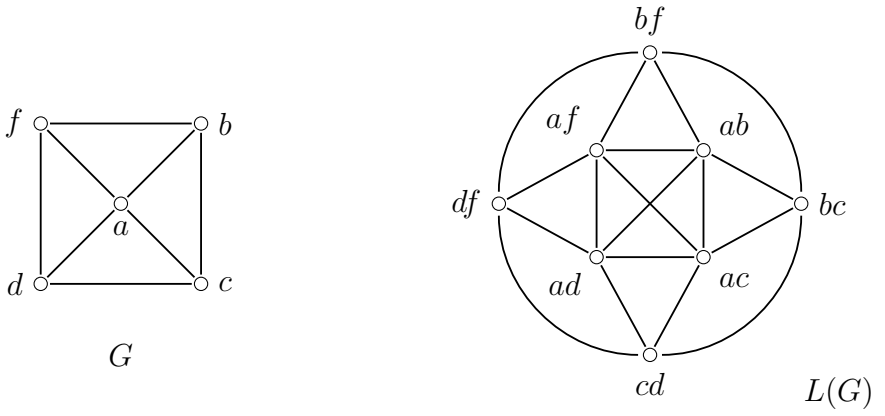


Figure 1.5: A graph G and its line graph $L(G)$.

1.3 Graph classes

We call a graph of order one a ***trivial*** graph; any other graph is called a ***non-trivial graph***. If a component of a graph G is a trivial graph with the only vertex v, we say that v is an ***isolated vertex*** of G, i.e., an isolated vertex of a graph is a vertex of degree zero. An ***empty graph*** is a graph whose edge set is empty, and a graph such that every pair of distinct vertices is adjacent is called a ***complete graph***. Thus, the complement of a complete graph is an empty graph, and vice versa. The only complete graph on n vertices is denoted by K_n. A non-empty vertex subset U of a graph G is called an ***independent set*** (or ***stable set***) if $G[U]$ is an empty graph, and it is called a ***clique*** if $G[U]$ is a complete graph. A clique C of a graph G is called ***maximal*** if no proper superset of C is a clique of G, and it is called ***maximum*** if there is no clique C' of G with $|C'| > |C|$. Analogous definitions are established for ***maximal independent sets*** and ***maximum independent sets***.

A graph $G = (V, E)$ is called an ***s-partite graph*** if there exists a ***partition***[1] $\{V_1, \ldots, V_s\}$ of V such that, for $i \in \{1, \ldots, s\}$, V_i is an independent set. If, in addition, $v_i v_j \in E$ for every $v_i \in V_i$ and $v_j \in V_j$, with $1 \leq i, j \leq s$, then we say that G is a ***complete s-partite graph***. We will say that a graph is a ***multipartite graph*** or a ***complete multipartite graph***, if it is, for some integer s, an s-partite graph or a complete s-partite graph, respectively. The 2-partite graphs receive the special name of ***bipartite graphs***; if $\{X, Y\}$ is a partition of $V(G)$, such that the ends of each edge of the graph G are one in X and the other in Y, then such a partition is called a ***bipartition*** of G, and the notation $G[X, Y]$ is normally used to refer that G is a bipartite graph with bipartition $\{X, Y\}$. A bipartite graph $G[X, Y]$ such that $xy \in E(G)$ for every $x \in X$ and every $y \in Y$ is called a ***complete bipartite graph***; the complete bipartite graph such that $|X| = a$ and $|Y| = b$ is usually denoted by $K_{a,b}$.

The complement of a complete k-partite graph is the disjoint union of at most k complete graphs, and it will be called a ***k-cluster***. In analogy of multipartite graphs, a ***cluster*** is a k-cluster for some integer k.

In the rest of this section, many different classes of graphs will be introduced. Although most of these classes are not essential for the development of the main results of this text, they are included so that it be self-contained. Some graph classes and concepts that are no introduced in this section are defined later in more appropriate places of the text, nevertheless, all the terms referred in this text are defined somewhere in it, and the reader can check the index in any moment to find any concept referred in this work.

A graph is said to be ***planar*** it if can be drawn in the plane so that its edges intersect only at their ends; such a drawing is called a ***planar embedding*** of the graph. A planar embedding of a graph G can be regarded as a graph isomorphic to G, for this reason, we often refer to a planar embedding of a planar graph as a ***plane graph***. A plane graph G partitions the plane into a number of arcwise-connected open sets that are called the ***faces*** of G; the degree of a face is the number of edges in its boundary. A plane graph in which all faces have degree three is sometimes called a *triangulation*; it is not difficult to prove that adding any edge to a triangulation result in a graph that is not planar, for this reason, *triangulations* are also known as ***maximal planar graphs***. A graph G is called ***outerplanar*** if it has a planar embedding in which all the vertices lie on the boundary of its only infinite face (its outer

[1] In the context of Graph Theory, a partition of a set A is understood as a finite set of **possibly-empty** disjoint subsets of A, such that their union is A.

face). The graph G on Figure 1.5 is planar, and it is not difficult to show that it is not outerplanar. On the other hand, the graph G of the Figure 1.3 is outerplanar, while the graph depicted in Figure 1.1 is the famous *Petersen graph*, which is not planar, and hence is neither outerplanar.

A graph G is ***biconnected*** if it is a connected graph and for any vertex v of G, $G-v$ is connected. A ***block*** (or ***biconnected component***) of a graph G is a maximal biconnected subgraph of G; a ***block graph*** is a graph in which every block is a complete graph (see Figure 1.6 for an example). Block graphs may be characterized as the intersection graphs of the blocks of arbitrary graphs.

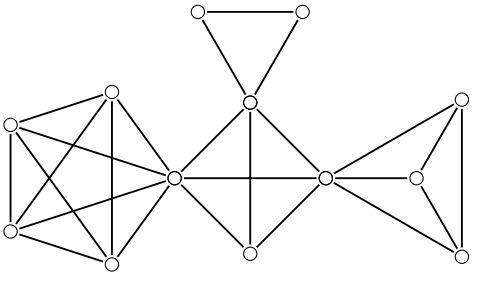


Figure 1.6: A block graph.

A graph G is called ***chordal*** if G does not contain any induced cycle of length greater than three. For a graph G, any chordal graph H with vertex set $V(H) = V(G)$ obtained by adding edges to G, is called a ***chordal completion*** of G, that is, H is a chordal completion of G if it is a chordal supergraph of G on the same vertex set. A chordal completion H of G is a ***minimal chordal completion*** of G if, for any $e \in E(H)$, the graph $H - e$ is not a chordal completion of G. Chordal completions and minimal chordal completions sometimes are also called ***triangulations*** and ***minimal triangulations***, respectively.[2]

The ***intersection graph*** G of a family of sets $S = \{S_1, S_2, \ldots, S_n\}$ is the graph with vertex set $V(G) = S$ and edge set $E(G) = \{S_iS_j \mid S_i \cap S_j \neq \varnothing\}$, i.e., G is such that two vertices are adjacent if they have a nonempty intersection; a graph G is an ***interval graph*** if G is the intersection graph of a family of intervals on the real line, or equivalently, if it is the intersection graph of connected subgraphs of a path graph. The ***comparablity graph*** of a partial

[2]Even though both of chordal completions and maximal planar graphs are commonly called triangulations, we will use the term *triangulations* only in the first sense.

order $\leq$ on a set V is the graph $G = (V, E)$ such that $uv \in E$ if and only if $u \leq v$ or $v \leq u$, that is, if u and v are comparable elements of V under $\leq$.

A ***permutation*** P of the set X is a sequence without repetition that includes all the elements of X. Given a permutation $P = (p_1, p_2, \ldots, p_n)$ on the set $\{1, 2, \ldots, n\}$, the ***permutation graph*** G associated with P is the graph with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$ and edge set

$$E(G) = \{v_i v_j \mid 1 \leq i < j \leq n \text{ and } p_i > p_j\}.$$

That is, the permutation graph G is the graph whose vertices represent the elements of the permutation P, and whose edges represent pairs of elements that are reversed by the permutation. Equivalently, permutation graphs can be defined geometrically as the intersection graphs of line segments in a *matching diagram*, where the permutation is represented putting its elements on two parallel lines (see Figure 1.7).

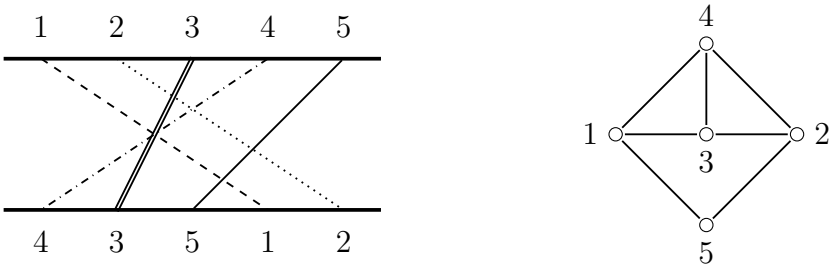


Figure 1.7: The matching diagram for the permutation $(4, 3, 5, 1, 2)$, and its corresponding permutation graph.

A ***trivially perfect graph*** is a graph G such that for every induced subgraph H of G, the size of a maximum independent set of H equals the number of maximal cliques of H [68]. A ***threshold graph*** is a graph that can be constructed from a trivial graph by repeated applications of the following two operations:

1. Disjoint union with a trivial graph.
2. Join with a trivial graph.

It can be shown that threshold graphs are a special case of trivially perfect graphs. A threshold graph is depicted in Figure 1.8.

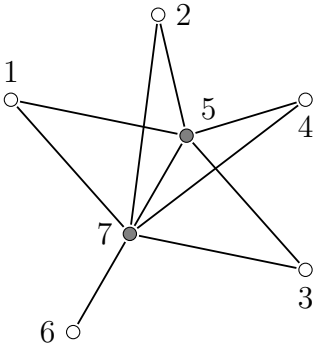


Figure 1.8: The graph shown is a threshold graph since it can be constructed begging with the single vertex graph whose only vertex is the vertex 1, and then adding the non-shaded vertices by means of disjoint unions and the shaded vertices by means of joins, in order in which they are numbered.

For any property $\mathcal{P}$ of graphs, we will say that a graph G is co-$\mathcal{P}$ if and only if $\overline{G}$ is a graph that satisfies property $\mathcal{P}$. For example, a graph G is said to be co-planar if $\overline{G}$ is a planar graph. Analogously, for a class of graphs $\mathcal{G}$, the class of graphs that are the complement of a graph in $\mathcal{G}$ is denoted by co-$\mathcal{G}$.

1.4 A very brief introduction to Complexity Theory

In this section the basic terminology concerning Complexity Theory is explained. Some concepts, as the meaning of *problem* or *algorithm*, are treated in a not-so-formal way because they are no fundamental on the core of this text. We are mainly focused on giving an intuitive idea on the way to measure the *quality* of an algorithm and the *difficulty* of a problem. Almost all of the content presented in this section was taken from [81]. Deeper treatment on this topic can be found in [62].

By an ***instance*** of a given problem we refer to the input data for the problem in a prescribed array; for example, for the problem of determining if a number is prime, an instance of the problem is a fixed number, as 10, 3, or any other. Intuitively, an ***algorithm*** is a technique which can be used to solve each instance of a prescribed problem. An algorithm is commonly described as

a set of rules that precisely define a sequence of operations in such a way that it allows us to obtain an ***output*** answer from the specific data of an ***input*** instance in a finite number of steps.

Complexity Theory studies the time (and memory space) an algorithm needs as a function of the size of the input data; this approach is used to compare different algorithms for solving the same problem. Concretely, the ***time complexity*** of an algorithm A is the function f such that $f(n)$ is the maximum number of *steps* that A needs to solve a problem instance having input data of length n. Note that we do not specify what a *step* really is, but usually are counted the basic arithmetic operations, access to arrays, comparisons, etc. each as one step.

In most cases it is impossible to calculate in a precise way the complexity of an algorithm, so we are satisfied with an asymptotic estimate of how fast it grows. To give such an estimate, we introduce the following notation. Let f and g be two functions from $\mathbb{N}$ to $\mathbb{R}^+$. We will say that

- f has at most rate of growth $g(n)$, denoted $f(n) = O(g(n))$, if there is a constant $c > 0$ such that $f(n) \leq cg(n)$ for all sufficiently large n;
- f has at least rate of growth $g(n)$, denoted $f(n) = \Omega(g(n))$, if there is a constant $c > 0$ such that $f(n) \geq cg(n)$ for all sufficiently large n;
- f has rate of growth $g(n)$, denoted $f(n) = \Theta(g(n))$, if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

An algorithm has complexity $O(g(n))$ if its time complexity is $O(g(n))$. For a given problem p, if there exists an algorithm having complexity $O(f(n))$, we say that p has complexity at most $O(f(n))$; if each algorithm for p has complexity $\Omega(g(n))$, then we say that p has complexity at least $\Omega(g(n))$, and, if in addition, there is an algorithm for p with complexity $O(g(n))$, then we say that p has complexity $\Theta(g(n))$.

The following sequence on growth rates is well known and can be found in [21]; in it is indicated that a function with complexity $O(n!)$ requires more computational time than a function with complexity $O(2^n)$, etc.

$$O(1) < O(\log n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n) < O(n!).$$

In practice, the ***polynomial algorithms*** (algorithms of complexity $O(n^k)$ for some k) have proved to be the most useful, for such a reason, these algorithms

are also called *efficient* or *good*. Problems for which a polynomial algorithm exists are also called *easy*, whereas problems for which no polynomial algorithm can exist are called ***intractable*** or *hard*.

1.4.1 NP-complete problems

Now we restrict our attention to ***decision problems***, that is, to problems whose solution is either *yes* or *no*. Within the class of decision problems, two great subclasses are of major importance: the class of all polynomial decision problems, that we will denote simply by P (for polynomial), and the class of decision problems such that each positive answer can be verified in polynomial time, which is denoted by NP (for non-deterministic polynomial); note that the definition of NP does not demand that a negative answer can be verified in polynomial time. Evidently $P \subseteq NP$, but it is not clear whether $P \neq NP$. In fact, the question whether $P = NP$ is an outstanding question of complexity theory.

A decision problem is called ***NP-complete*** if it is in NP and if the polynomial solvability of this problem would imply that all other problems in NP are also solvable in polynomial time. A decision problem which is not necessarily in NP, but whose polynomial solvability would nevertheless imply P=NP is called ***NP-hard***. Observe that if we could find a polynomial algorithm for an NP-complete problem, we would prove that $P = NP$.

Although there was not a clear reason why any NP-complete problem should exist, in 1971 Stephen Cook showed the existence of these problems by proving that the known SAT-problem (or boolean satisfiability problem) is one of them [32]. Once a problem is known to be NP-complete, other problems can be shown to be NP-complete by *transforming* the NP-complete problem in polynomial time to these problems. Concretely, if we wish to prove that a problem Q in NP is NP-complete, it is enough to show that, for an NP-complete problem Q', there exists a polynomial algorithm that assigns to each instance of Q' an instance of Q in such a way that an instance of Q' has solution "yes" if and only if its corresponding output also has solution "yes". An algorithm as described before is called a ***polynomial-time reduction*** from Q' to Q. It is worth mentioning that, only one year later that the existence of NP-complete problems was exhibited, it was published the famous *Karp's 21 NP-complete problems* on combinatorics and graph theory [82]. Unlike the SAT-problem, that required of particular techniques to prove that it is NP-complete, the 21 problems exposed by Karp were proved to be NP-complete by the technique of

polynomial-time reductions.

Hundreds of problems have been recognized as NP-complete, but for none of these problems a polynomial algorithm could be found in spite of enormous efforts, which gives support for the conjecture P $\neq$ NP.

1.4.2 Certifying algorithms

A ***certifying algorithm*** is an algorithm that outputs, together with a solution to the problem it solves, a proof that the solution is correct; such a proof is called a ***certificate***. Particularly, for an algorithm whose possible output are only *yes* or *no*, called ***decision algorithms***, a certificate of a yes-output (no-output) is called a ***yes-certificate*** (***no-certificate***).

Whenever an algorithm is run, one of three things happens: it produces a correct output (the desired case), it is detected a bug in the algorithm (undesired, but generally preferable to continuing without detecting the bug), or the algorithm fails in a way that masks bugs and prevents it from being detected (completely undesired). Certificates prevent the latter case occur. That is why implementations of certifying algorithms (including a checker for the certificates) may be considered to be more reliable than non-certifying algorithms.

The proof produced by a certifying algorithm should be in some sense simpler than the algorithm itself, for otherwise any algorithm could be considered certifying (with its output verified by running the same algorithm again). Sometimes this is formalized by requiring that a verification of the proof take less time than the original algorithm.

Chapter 2

Polar graphs

2.1 Colorings

A ***k-vertex coloring***, or ***k-coloring*** in short, of a graph G is a function c from the vertex set of G into a set S of cardinality k, traditionally named the set of colors and usually taken as the set $\{1, 2, \ldots, k\}$; thus, a k-coloring of G is merely an assignment of k colors to the vertices of G. It is understood that a ***coloring*** of G is a k-coloring of G for some positive integer k. A coloring c of G is a ***proper coloring*** if the images under c of adjacent vertices are distinct, in other words, a coloring is proper if no two adjacent vertices are assigned the same color. Figure 2.1 shows an example of a proper coloring of a graph.

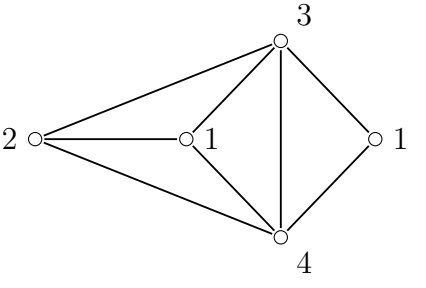


Figure 2.1: A 4-proper coloring of a graph.

Given a graph G and a k-coloring c of G, the preimage of each color under c is called the ***color class*** corresponding to that color. Observe that the set of the preimages of a coloring of G is a partition of the vertex set of G (with some

parts possibly empty). At the same time, given a partition of the vertex set of a graph G, there exists an natural coloring of G with so many colors as parts in the partition. For this reason colorings and vertex partitions are commonly considered as the same. Note also that a coloring is proper if and only if every color class is an independent set of the colored graph; this property will be important later, when we talk about coloring generalizations.

The minimization problem for vertex coloring

A graph G is k-***colorable*** if there exists a k-proper coloring of G. The minimum k for which a graph G is k-colorable is the ***chromatic number*** of G, and it is denoted by $\chi(G)$, or simply χ if there is no risk of confusion. Finally, G is a k-***chromatic*** graph if $\chi(G) = k$. For example, the graph depicted in Figure 2.1 is not only 4-colorable, its chromatic number is four since it contains K_4 as a subgraph.

Coloring applications arise in day-to-day situations where it is required to partition a set of objects into groups in such a way that the members of each group are *compatible* according to some preset criterion. We give below two typical examples of such problems.

Suppose that a chemistry laboratory wants to store its chemicals, but some chemicals may cause violent reactions if they are brought together. For a safe storage, incompatible chemicals should be stored in distinct rooms of its warehouse. Of course that an easy way to accomplish this is storing only one chemical by room, but this is certainly not the best way of doing it because we should be using many more rooms than we really need. So a natural question is, what is the minimum number of rooms required to store all the chemicals so that in each room only compatible chemicals are stored?

Our *chemical problem* can be easily modeled by graphs as follows. Let G be the graph whose vertex set is the set of the chemicals, and let two vertices be adjacent if and only if the corresponding chemicals are incompatible. Observe that a proper coloring of G induces a partition of the vertex set of G such that each chromatic class is a collection of compatible chemicals, and vice versa, a partition of the vertex set of G into independent sets induces naturally a proper coloring of G with so many colors as the elements in the partition. The question about the minimum number of rooms required to store the chemicals in a safe mode is thus converted into the question, what is the minimum number k such that exists a k-proper coloring of G?, or what is the same, what is the chromatic number of G?

For our second example lets consider the *register allocation problem*. In many programming languages, the programmer may use any number of variables. The computer can quickly read and write registers (the ideal locations of variables) in the CPU, so the computer program runs faster when more variables can be in the CPU's registers. However, the number of registers is limited in most CPUs. Therefore, when the compiler is translating code to machine-language, it must decide how to allocate variables to the limited number of registers in the CPU.

Not all the variables are in use (or "live", as is said) at the same time, so over the lifetime of a program a given register may be used to hold different variables. However, two variables in use at the same time cannot be assigned to the same register without corrupting one of the variables.

Compilers can determine which sets of variables are live at the same time. Using this information, an *interference graph* can be construct in such a way that the vertices of the graph are the variables of the program, and the *interference edges* connect pairs of vertices which are live at the same time. *Register allocation* can then be reduced to the problem of k-coloring the resulting graph: if the interference graph is k-colorable, then there is a register assignment that uses no more than k-registers.

Brooks' theorem

If H is a subgraph of G and G is a k-colorable graph, then so is H. Thus $\chi(G) \geq \chi(H)$ for every subgraph H of G. In particular, if G contains a copy of the complete graph K_r, then $\chi(G) \geq r$. Therefore, naming ***clique number*** to the largest r such that G contains a copy of the complete graph K_r, and denoting it by $\omega(G)$, we have that for any graph G, $\omega(G) \leq \chi(G)$, that is, the clique number is always a lower bound for the chromatic number.

An upper bound for the chromatic number of a graph G can be obtained by means of a ***greedy coloring*** heuristic: Consider a linear order of the vertices of G, let say $v_1, \ldots, v_n$. Assign the color 1 to the vertex v_1 and subsequently, for $2 \leq i \leq n$, color the vertex v_i with the first positive integer that has not been assigned to its colored neighbors. A coloring obtained by this way is evidently a proper coloring of G. Moreover, in a greedy coloring of G every vertex receive a color in the set $\{1, 2, \ldots, \Delta(G) + 1\}$, and therefore, for every graph G, the relation $\chi(G) \leq \Delta(G) + 1$ is satisfied.

Although the latter bound for the chromatic number is best possible, being attained by odd cycles and complete graphs, Brooks showed in 1941 that these

are the only connected graphs for which equality holds.

Theorem 2.1 (Brooks' Theorem, 1941)**.** *[18, 88]*
Let G be a connected graph other than an odd cycle or a complete graph. Then $\chi(G) \leq \Delta(G)$.

2.1.1 Perfect graphs

As we mentioned above, in every graph the clique number provide a lower bound for the chromatic number. Then, like Brooks' Theorem identify the graphs that attain the upper bound of the chromatic number, $\chi \leq \Delta + 1$, one can ask about the graphs that attain the lower bound $\chi \geq \omega$.

However, this question is not particularly interesting because there exist many examples of graphs G for which $\chi(G) = \omega(G)$, such as complete graphs, bipartite graphs, and any disjoint union of the complete graph K_k with a k-colorable graph. A more interesting thing occurs if one request that the equality $\chi = \omega$ hold not only for G, but also for all of its induced subgraphs.

Berge define the ***perfect graphs*** as the graphs G such that $\chi(H) = \omega(H)$ for every induced subgraph H of G [5]. He observed that the graphs satisfying this property include many basic families of graphs, such as bipartite graphs and its line graphs, chordal graphs (that include split graphs[1], block graphs, interval graphs, trivially perfect graphs and threshold graphs), and comparability graphs (that include complement of interval graphs and P_4-free graphs).

The perfect graph theorems

Around 1960[2], during the initial work of Berge on perfect graphs, he made two important conjectures on the structure of these graphs [87]. One of these conjectures was based on a Berge's sharp observation: the complement of every perfect graph is also a perfect graph. Berge conjectured that the converse was also true, and this affirmation was verified by Lovász in 1972, resulting in what is known as the Perfect Graph Theorem.

Theorem 2.2 (Perfect Graph Theorem, 1972)**.** *[87]*
A graph is perfect if and only if its complement is perfect.

[1] Graphs whose vertex set can be partitioned into a clique and a stable set

[2] According to [6] at least one of these conjectures was expressed in 1960, while according to [8] both conjectures dates from 1963, and according to [22] both of them dates from 1961.

Thus, graph perfection, defined as the equality of clique number and chromatic number in every induced subgraph, is equivalent to the equality of ***independence number*** (maximum independent set size) and ***clique covering number*** (the minimum number of cliques needed to cover the vertex set).

An interesting characterization of perfect graphs, stronger than the Perfect Graph Theorem, was proposed by A. Hajnal shortly thereafter that the proof of the latter was given. This new characterization was also confirmed by Lovász, too in 1972.

Theorem 2.3 (Lovász, 1972)**.** *[86]*
A graph G is perfect if and only if every induced subgraph H of G satisfies the inequality

$$|V(H)| \leq \omega(\overline{H})\omega(H). \tag{2.1}$$

Note that Theorem 2.3 implies the Perfect Graph Theorem since the inequality (2.1) is invariant under complementation.

If a graph is perfect, then so are all of its induced subgraphs. This means that one can characterize the perfect graphs by describing all minimally imperfect graphs (graphs that are not perfect but every of its proper induced subgraphs are perfect), that is, one can give a forbidden graph characterization of perfect graphs. Some easy examples of minimally imperfect graphs can be given, for instance, every odd cycle of length at least five: its chromatic number is 3 while its clique number is 2, so an odd cycle of length at least 5 is not a perfect graph. Moreover, every proper induced subgraph of these cycles are bipartite graphs and in consequence they are perfect graphs. Therefore every odd cycle of length greater than 3 is a minimally imperfect graph. Observe that as a consequence of the Perfect Graph Theorem (although it can be proved directly without difficulty), the complement of every odd cycle of length at least 5 is also a minimally imperfect graph. An induced cycle of odd length at least five is called an ***odd hole***, while an induced subgraph that is the complement of an odd hole is called an ***odd anti-hole***.

In addition to the Perfect Graph Theorem, the other conjecture about perfect graphs given by Berge in 1960's was that the only minimal imperfect graphs are the odd cycles of length at least five and its complements, it is to say, that a graph is perfect if and only if it contains neither odd holes of length greater than 3 nor odd anti-holes of length greater than, 3 as induced subgraphs. Berge named this conjecture the Strong Perfect Graph Conjecture because its truth would imply the Perfect Graph Theorem. In 2006, some forty

years later than Berge gave his conjecture, a proof of it was finally published by Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas.

Theorem 2.4 (Strong Perfect Graph Theorem, 2006)**.** *[26]*
A graph is perfect if and only if it contains no odd cycle of length at least five, or its complement, as an induce subgraph.

A polynomial time recognition algorithm for graphs without odd holes of length at least five as induced subgraphs, nor its complements (customarily called Berge graphs), was given by Chudnovsky et al. [25] shortly after of that the proof of the Strong Perfect Theorem was obtained, but independently to the latter. Thus, through the Strong Perfect Graph Theorem, this algorithm results also in a polynomial time recognition algorithm for perfect graphs.

2.1.2 Coloring algorithms on perfect graphs

As we mentioned above, a greedy coloring of a graph is a proper coloring of its vertices obtained by a greedy algorithm which considers the vertices of the graph in a preset sequence and assigns to each vertex the first available color. The maximum number of colors that can be used by a greedy coloring of a graph G is called its ***Grundy number***, and it is usually denoted by $\Gamma(G)$. As is evident, a greedy coloring of a graph G cannot use less colors than the chromatic number of G, so that for every graph G, $\chi(G) \leq \Gamma(G)$.

Greedy colorings do not in general use the minimum number of colors possible, in fact, a greedy coloring can use many more colors than an optimal proper coloring. A typical example of it is the ***crown graph***, that is a complete bipartite graph $K_{n,n}$ with the edges of a perfect matching removed: this graph, being bipartite, has chromatic number two, while there exists an ordering of its vertices that produce a greedy coloring with n colors (see Figure 2.2).

Example above shows that the vertex ordering in a greedy coloring is a very important factor, so much so that, although the difference between the chromatic number and the Grundy number of a graph can be arbitrarily large, it always exists an order of the vertices in such a way that the corresponding greedy algorithm produces an optimal coloring: given any optimal coloring of a graph G with colors $1, 2, \ldots, k$ in which the color class corresponding to color 1 is maximal, the color class corresponding to color 2 is maximal with respect to the color 1, an so on, one may order the vertices by their colors (first all the vertices in the 1-color class, then all the vertices on the 2-color

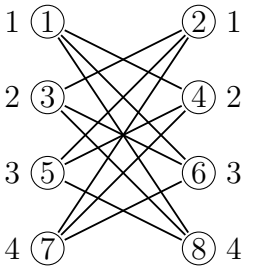


Figure 2.2: A greedy coloring of the crown graph on eight vertices that requires four colors. Numbers inside the vertices indicate the order used for the greedy coloring, while numbers next to vertices indicate the color assigned.

class, etc.). Evidently, when one uses a greedy algorithm with this order, the resulting coloring is optimal.

It is NP-complete to decide if a given graph admits a k-coloring for a given k except when $k \leq 2$; in particular, it is NP-hard to compute the chromatic number of a graph [62], so that, we have in one hand that graph coloring is computationally hard in general. On the other hand, a greedy coloring algorithm can be implemented in linear time $O(n + m)$, but as we have mentioned above, greedy colorings can be arbitrarily bad. Nevertheless, as we will see below, in some subclasses of perfect graphs, greedy coloring algorithms can be used for finding optimal colorings.

Although no efficient combinatorial algorithms are known for coloring perfect graphs[3], for some interesting subfamilies of perfect graphs such algorithms are well-known. Some results on coloring for classical subclasses of perfect graphs are presented below.

A ***perfect elimination ordering*** of a graph $G = (V, E)$ is a linear order $\leq$ of V such that, for every vertex $v \in V$, the set

$$\{w \in V \mid v \leq w,\ w \in N(v) \cup \{v\}\}$$

is a clique of G, that is to say, for each vertex $v \in V$, v and the neighbors of v that are greater than v in the order $\leq$ form a clique of G.

Knowing of a perfect elimination ordering $\leq$ for a given graph G can be used to coloring it optimally. Because, if a greedy coloring algorithm is applied

[3]Grötschel, Lovász and Schrijver [70] proved the existence of a polynomial time algorithm for solving the coloring problem on all perfect graphs. Nevertheless, this algorithm use the ellipsoidal method for linear programming and is not useful in the practical.

to G using the reverse order of $\leq$, the resulting coloring uses trivially only $\omega(G)$ colors, and therefore it is optimal (see Figure 2.3 for an example).

Figure 2.3: As can be checked, the numbers inside the vertices indicate a perfect elimination ordering of the path on four vertices. Then, using the reverse order in a greedy coloring algorithm, it is obtained an optimal coloring for this graph, which is indicated by the numbers above the vertices. Notice that if we apply a greedy coloring algorithm using this perfect elimination ordering, instead of its reverse, the resulting coloring is no longer optimal.

Notice that the argument above on optimal coloring of graphs having a perfect elimination order also shows that these graphs satisfy the equation $\chi = \omega$. In addition, it is not difficult to prove that perfectly orderable graphs are exactly the chordal graphs [60]. Then, by the previous comments, and since chordal graphs are a hereditary class of graphs (i.e., every induced subgraph of a chordal graph is a chordal graph), we have an easy proof that chordal graphs are perfect[4].

Rose, Lueker and Tarjan gave an algorithm based on lexicographic breadth-first search for finding a perfect elimination ordering in a graph, if one exist, in $O(n + m)$ time [97]. Then, chordal graphs can be efficiently recognized and colored.

It is not difficult to observe from its definition that threshold graphs admit a perfect elimination ordering, and therefore they are chordal graphs. Additionally, it can be shown that a graph is chordal if and only if it is the intersection graph of connected subgraphs of a tree, and since interval graphs are intersection graphs of connected subgraphs of paths, we have that interval graphs are chordal graphs too. Thus, linear algorithms for coloring chordal graphs also work for interval and threshold graphs.

Chordal graphs are special cases of a wider subclass of perfect graphs, the so-called perfectly orderable graphs. A graph G is said to be ***perfectly orderable*** if its vertex set can be ordered in such a way that a greedy coloring algorithm with that ordering optimally colors every induced subgraph of G;

[4]Another short proof can be easily given taking into account the Strong Perfect Graph Theorem.

such an ordering is called a ***perfect ordering*** of G. Perfectly orderable graphs were proved to be perfect graphs in [31]. Unfortunately, recognizing perfectly orderable graphs is NP-complete, and finding a perfect ordering of a graph, even if the graph is known to be perfectly orderable, is NP-hard [92].

As we said before, chordal graphs are perfectly orderable, because a perfect ordering of a chordal graph may be found by reversing a perfect elimination ordering of the graph. In addition to the chordal graphs, other interesting graphs are also perfectly orderable, such is the case of comparability graphs and distance-hereditary graphs (described below) [12].

Comparability graphs have been characterized as the graphs that have a transitive orientation [63], and a perfect ordering for a comparability graph is given by an ordering of its vertices in such a way that, for a transitive orientation of the graph, vertex u comes before vertex v in the ordering whenever there is an arc from u to v. Comparability graph recognition and finding a transitive orientation can be done in $O(\Delta m)$ time [67]. Co-interval, permutation, bipartite, trivially perfect and threshold, are all subclasses of comparability graphs.

Distance-hereditary graphs are the graphs for which the distances in any connected induced subgraph are the same as they are in the original graph, formally, a graph G is a distance-hereditary graph if for any two of its vertices, u and v, if both of them belong to a connected induced subgraph H of G, then the distance between u and v in G is the same that the one in H. Distance-hereditary graphs can be characterized in several equivalent ways [77], and they can be recognized in $O(n + m)$ time [4]. As we mentioned before, every distance-hereditary graph is a perfect graph [77], and more specifically a perfectly orderable graph [12]

A very special subclass of perfect graphs are the P_4-free graphs, also called ***complement reducible graphs***, or ***cographs*** in short. Cographs are a subclass of both comparability graphs and distance-hereditary graphs, and then for the perfectly orderable graphs.

Moreover, a simple characterization of perfect orderings given for Chvátal establish that a linear vertex ordering $<$ of a graph G is perfect if and only if there is no induced P_4, (a, b, c, d), such that $a < b$ and $d < c$ [31]. Because any cograph is P_4-free, none of its vertex orderings can violate the path-ordering required on a perfect ordering, and so, every vertex ordering of a cograph is a perfect order.

Much more will be said about cographs in Chapter 3, which is specially devoted to this class of graphs.

2.2 Coloring generalizations

2.2.1 Cocolorings

We observed before that a proper coloring of a graph G corresponds to a partition of the vertex set of G into independent sets. Thus, given the dual relation between independent sets and cliques, a natural variant of proper colorings could be replacing partitions into independent sets by partitions into cliques, but this just corresponds to proper colorings of the graph complement. A more interesting possibility about extending the concept of proper coloring is described below.

A ***cocoloring*** of a graph G is a coloring of G such that each color class is an independent set or a clique in G. Equivalently, a cocoloring of G is an assignment of colors to the vertices of G such that every color class is an independent set in G or in $\overline{G}$. The ***cochromatic number*** of G, denoted by $z(G)$, is defined as the minimum possible number of color classes in a cocoloring of G. Observe that for every graph G, any proper coloring of G is also a cocoloring of G. It follows that the cochromatic number of G is less than or equal to the chromatic number, that is, $z(G) \leq \chi(G)$ for every graph G.

Cocolorings were named and first studied by Lesniak and Straight in 1977 [85]. Subsequent papers about cocolorings studied the structures of critical cochromatics graphs [16, 64, 66], bounds on the cochromatic numbers of graphs with certain properties [1, 49, 50, 51, 64, 100, 101], and complexity of the cocoloring problem in some graph classes including chordal graphs and cographs [65].

(k, l)-colorings

One can think about finding cocolorings of a graph restricting the number of color classes that are permitted to be independent sets or cliques; this is analogous to ask if a graph admits a proper k-coloring for a fixed k. This idea gives origin to the concept of (k, l)-colorings.

A (k, l)***-coloring*** of a graph G is a cocoloring of G such that at most k of the color classes are independent sets and at most l of the color classes are cliques. For k and l fixed, if a (k, l)-coloring of G exists, then we say that G is a (k, l)***-colorable graph***, or a (k, l)***-graph*** in short. Thus, a $(k, 0)$-coloring is simply a proper k-coloring and a $(k, 0)$-colorable graph is the same that a k-colorable graph.

The $(1,1)$-graphs are the graphs whose vertex set can be partitioned into a clique and an independent set; these graphs are called ***split graphs*** and they have been widely studied along the years. Splits graphs are a very special subclass of perfect graphs, and they admit efficient algorithms for many standard combinatorial optimization problems that are NP-complete on more general graph families, as the graph coloring problem [68].

However, although split graphs present an excellent behavior, for most values of k and l, (k,l)-graphs do not. In fact, only when $k \leq 2$ and $l \leq 2$, (k,l)-graphs can be recognized in polynomial time [10, 11, 13, 56], while that when k or l is at least 3 it is NP-complete to recognize (k,l)-graphs [10, 13].

Results about the (k,l)-coloring problem for general graphs have been obtained by Feder et al. [56], for chordal graphs by Hell et al. [75], for some special classes of graphs as cacti in the case $k = l$ by Ekim and de Werra [42], and for cographs by Demange et al. [40].

2.2.2 Matrix partitions

Many problems in graph theory can be established in terms of partitions of the vertex set of a graph G [56]. In a lot of cases these problems seek for a partition into subsets satisfying some of the following conditions: first, may be required that every pair of distinct vertices in a set of the partition is adjacent (the part is a clique) or that none pair of vertices is adjacent (the part is an independent set), and second, may be required that some pair of the elements of the partition is completely adjacent (each vertex of a part is adjacent to each vertex of the other) or completely non-adjacent (no vertex of one part is adjacent to any vertex of the other).

In [55, 56] it was introduced a common generalization of such problems as follows: partition the vertices of an input graph into k parts $V_1, V_2, \ldots, V_k$ with a fixed 'pattern' of requirements as to which $V_i's$ are independent sets or cliques, and which pairs V_i, V_j are completely adjacent or completely non-adjacent. These requirements may be conveniently captured by a symmetric $k \times k$ matrix M in which the diagonal entries $M_{i,i}$ encode the internal restriction on the sets V_i, and the off-diagonal entries $M_{i,j}$, $i \neq j$, encode the restriction on the edges between V_i and V_j.

The previous paragraph can be rewritten as follows. Let M be a symmetric $m \times m$ matrix over the symbols $0, 1$ and $*$. An ***M-partition*** of a graph G is a partition of the vertices of G into sets $V_1, V_2, \ldots, V_m$ such that two distinct vertices in parts V_i and V_j are adjacent if $M_{i,j} = 1$, non adjacent if $M_{i,j} = 0$,

and they have no adjacency restriction if $M_{i,j} = *$. Note that when $i = j$ these restrictions mean that part V_i is either a clique, or an independent set, or it is unrestricted, when $M_{i,i}$ is 1, or 0, or $*$, respectively.

Observe that if M has an entry $*$ in its diagonal, then every graph has an M-partition because all vertices can be placed in the corresponding unrestricted part. For this reason, it is customary to focus on matrices M with only zeros and ones in its main diagonal; in fact, we assume that each matrix M consist of two diagonal block matrices, A and B, and the off-diagonal block matrices C and C^T, where A is a $k \times k$ matrix with zeros on its diagonal, B is an $l \times l$ matrix with a diagonal of ones, and C is a $k \times l$ matrix (see (2.2)). In the following, we will talk of the A, B and C blocks of a matrix M for referring the block matrices before described.

$$M = \left(\begin{array}{c|c} A & C \\ \hline C^T & B \end{array}\right). \tag{2.2}$$

For a fixed matrix M, the ***M-partition problem*** asks whether or not an input graph G admits an M-partition; if G admits an M-partition we say that G is an ***M-partitionable graph***. On the other hand, if a graph G fails to admit an M-partition, we say that G is an ***M-obstruction***, and if in addition the deleting of any vertex of G results in an M-partitionable graph, then G is a ***minimal M-obstruction***. We remark that some sets in an M-partition may be empty.

Observe that a k-coloring corresponds to an M-partition where M is the $k \times k$ matrix with zeros in its main diagonal and ones in any other place. Also cocolorings are generalized by M-partitions: a symmetric matrix M of size $(k+l) \times (k+l)$ with k zeros and l ones in its main diagonal, and $*$ in any other entry corresponds to a (k, l)-cocoloring.

But cocolorings are not the unique generalization of colorings that are included in the M-partitions. A homomorphism of a graph G into another graph H is a mapping $f : V(G) \to V(H)$ such that $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$. Since a homomorphism $f : G \to K_k$ is simply a k-proper coloring of G, the concept of homomorphism can be thought as a generalization of the vertex coloring, reason why the term H-coloring of G has been used to describe a homomorphism of G into H. Furthermore, if a matrix M is obtained from the adjacency matrix of a graph H by replacing each 1 by an $*$, then each H-coloring f of G corresponds to an M-partition of G, where the parts are the

sets $f^{-1}(h)$ for $h \in V(H)$. Thus M-partitions also generalize H-colorings.

In following section will introduce another kind of graph partitions that are also a special case of matrix partitions. A deeper development on the topic of matrix partitions can be found in [73].

2.3 Polar graphs

Although split graphs were first studied by Földes and Hammer in 1976 [58], these graphs were also independently introduced a few years later by Chernyak and Tyshkevich, who additionally gave in 1985 a generalization of such family of graphs which they named *polar graphs* [104].

In current terms, a graph $G = (V, E)$ is a ***polar graph*** if V admits a partition $\{A, B\}$ such that A induces a complete multipartite graph and B induces a ***cluster***, that is to say, a disjoint union of complete graphs with no other edges; the partition defined above is called a ***polar partition*** of G. Notice that since a cluster is the complement of a complete multipartite graph, it is immediate from the definition that the complement of a polar graph is also a polar graph. Other ways to say that a graph G is a polar graph is to say that there exists a partition $\{A, B\}$ of $V(G)$ such that all connected components of the graphs $\overline{G[A]}$ and $G[B]$ are complete graphs, or say that $G[A]$ is $\overline{P_3}$-free and $G[B]$ is P_3-free.

Observe that not every graph is polar, for example, as we explain below, the graph G on the top of Figure 2.4 is not a polar graph. Notice that every polar partition $\{A, B\}$ of G, restricted to the 6-cycle $C = (v_1, v_2, \ldots, v_6, v_1)$, is a polar partition of that cycle, and it is straightforward to show that there are essentially three polar partitions of C, which are shown on the bottom of Figure 2.4. Nevertheless, none of the polar partitions of C can be extended to a polar partition of G: for the first partition shown in Figure 2.4, neither the vertex y nor the vertex z could be added to the A part, or $G[A]$ would have an induced $\overline{P_3}$, but if both of y and z are added to the part B, then $G[B]$ would have an induced P_3, which cannot be; similarly, in the remaining cases, the vertex z cannot be added neither to the part A nor the part B. Thus, G is not a polar graph.

The graph G shown in Figure 2.4, besides not being a polar graph, has the property that if any vertex is removed, the remaining graph is polar. The claim above can be verified in Figure 2.5, where polar partitions of $G - v$ are shown for each vertex v of G. A graph G that is not a polar graph is called a ***polar***

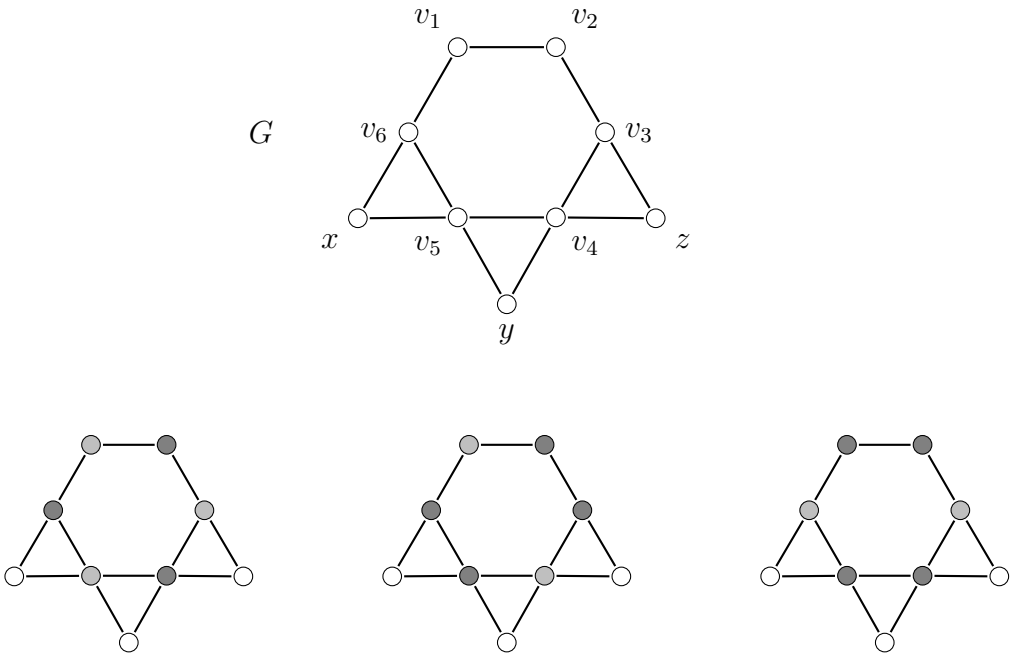


Figure 2.4: In the top, a graph G that is not a polar graph. In the bottom, the three possible polar partitions of the cycle $C = (v_1, v_2, \ldots, v_6, v_1)$; the dark-gray shaded vertices induce in each case a cluster, while the light-gray shaded vertices induce empty graphs, which are complete multipartite graphs.

obstruction, and a polar obstruction such that, for every vertex v of G, the graph $G - v$ is a polar graph is called a ***minimal polar obstruction***. Thus, the graph G shown in Figure 2.4 is a minimal polar obstruction.

Since an empty graph is merely a complete 1-partite graph and a complete graph is a trivial union of complete graphs, it is evidently that polar graphs are a natural generalization of split graphs. But split graphs are not the only interesting subfamily of polar graphs, bipartite graphs and co-bipartite graphs are also polar graphs: if G is a bipartite graph with parts X and Y, then $\{X, Y\}$ is a polar partition for both G and $\overline{G}$.

It is well known that bipartite graphs can be recognized in time $O(n + m)$ by means of a Breadth-First Search algorithm, and since the complement of a graph can be found in linear time $O(n^2)$, co-bipartite graphs also can be recognized efficiently. In fact, Joeris, Lindzey, McConnell and Osheim provided

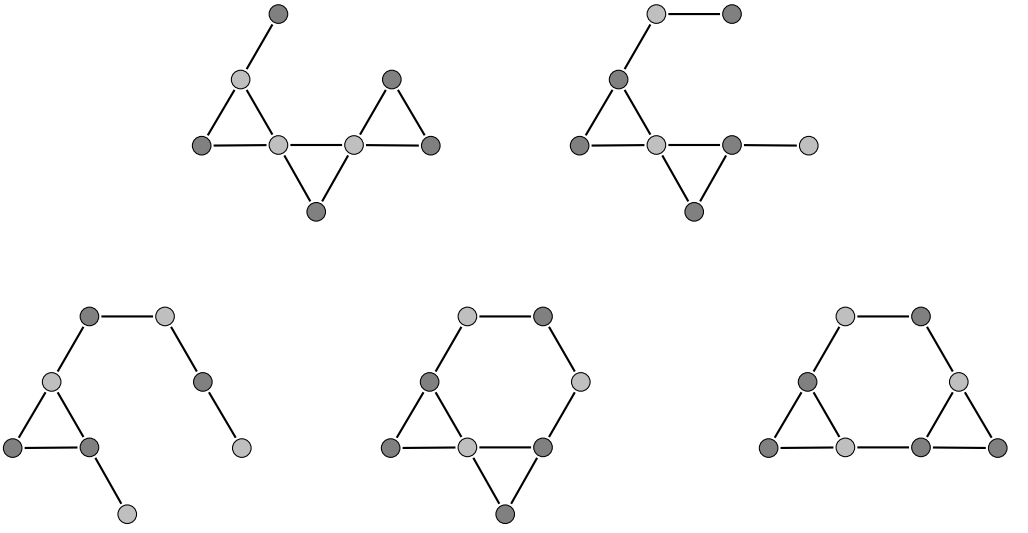


Figure 2.5: Polar partitions for the one-vertex-deleted subgraphs of the non-polar graph G shown in Figure 2.4.

in [79] of a Depth-First Search algorithm in linear time $O(n + \overline{m})$ for the complement of a graph G, where $\overline{m}$ stands for the size of $\overline{G}$; such an algorithm also can be used for recognizing co-bipartite graphs in linear time.

Splits graphs were characterized as the family of $\{2K_2, C_4, C_5\}$-free graphs, so they can be recognized in polynomial time [58]. Moreover, Hammer and Simeone gave a characterization of split graphs using only degree sequences, and they derived from it a linear time recognition algorithm for split graphs [71], and Hell, Klein, Noguiera and Protti [75] showed that split graphs can be recognized in linear time even if some vertices are preassigned to be in the stable or in the clique set. Regrettably, polar graphs are not so easy to recognize, A. A. Chernyak and Z. A. Chernyak showed that the problem of deciding whether a graph is a polar graph is an NP-complete problem [23].

In [104] the (α, β) class of polar graphs was defined for every pair of positive integers α and β. A graph G belongs to the (α, β) class (of polar graphs) if G admits a polar partition $\{A, B\}$ such that the order of each connected component of $\overline{G[A]}$ and $G[B]$ do not exceed α and β, respectively. For convenience, we will say that a graph G belongs to the (α, ∞) class if, for some positive integer β, G belongs to the (α, β) class of polar graphs; that is to say, a graph G belongs to the (α, ∞) class of polar graphs if G admits a polar partition $\{A, B\}$ such

that the order of each connected component of $\overline{G[A]}$ is at most α, without restrictions on $G[B]$. The (∞, β) class of polar graphs is defined analogously. Observe that a graph G is a split graph if and only if G belongs to the $(1, 1)$ class.

Mel'nikov and Kozhich [90] solved, for fixed positive integers α and β, the problem of recognizing the graphs of the (α, β) class with a $O(n^{2\alpha+2\beta+3})$ algorithm, where n is the order of the graph. Furthermore, Tyschkevich and Chernyak [105] showed that the problem of recognizing graphs in the $(1, \beta)$ class can be solved in time $O(n^3)$ for $\beta \in \mathbb{N} \setminus \{0\}$, and even for the case (α, ∞), while Chernyak et al. [23] proved that for $\beta > 1$ deciding if a graph is in the (∞, β) class is an NP-complete problem.

Special classes of polar graphs

Since recognizing polar graphs turned out to be an NP-complete problem in general, it has been of interest restricting the problem to special classes of graphs, looking for some kinds of them with a significant impact on the complexity. For a count of the results in this line of investigation, some new concepts will be useful.

Let $G = (V, E)$ be a graph. Then, G is called ***monopolar*** if V admits a polar partition $\{A, B\}$ such that A is an stable set, and it is called ***unipolar*** if V admits a polar partition $\{A, B\}$ such that A is a clique (i.e., G is in the $(1, \infty)$ class). On one hand, first results on unipolar graphs show that they can be recognized in $O(n^3)$ time [105]. Recently, it has been shown that, in fact, unipolar graphs can be recognized in $O(nm + nm_f)$ time, where m_f is the number of edges added in a minimal triangulation of a given graph [52], or alternatively, in $O(n^2 m)$ time [30]. On the other hand, recognizing monopolar graphs is NP-complete in general [54]. These two subclasses are interesting because most of the known recognition algorithms for polar graphs of a specific graph class first check unipolarity or monopolarity and then, if both are not true, test if there is a polar partition. Known results on the complexity of recognition on subfamilies of polar graphs are summarized below.

In [47] Ekim, Mahadev and De Werra provided a characterization of polar cographs by means of a finite family of forbidden subgraphs; they also proved a characterization of cographs that are or monopolar or complement of a monopolar, based on a family of 16 forbidden subgraphs [48]. Besides, Ekim et al. gave a linear time algorithm for finding the largest induced polar subgraph in cographs, which also serve as a (mono)polar cograph recognition algorithm.

Ekim, Hell, Stacho and De Werra [45] offered a polynomial time algorithm for finding a polar partition of a chordal graph, if one exists. Besides, they obtained an $O(n^3)$ unipolarity algorithm for chordal graphs and a linear time algorithm to decide whether a chordal graph is monopolar, even in the list version of the problem; the last algorithm also finds a monopolar partition if one exists. It is worth noting that unipolar chordal graphs were characterized several years before as chordal $2P_3$-free graphs by Gagarin [61].

Unlike the results obtained for polar cographs [47], it was shown in [45] that for chordal graphs there are infinitely many minimal non-polar and non-monopolar graphs; nevertheless, it was proved in [99] that all chordal minimal non-monopolar graphs can be generated by a simple recursive procedure.

Ekim, Heggernes and Meister [43] worked with graphs that are both polar and permutation graphs, solving a question that had been suggested in [47]. They showed that for permutation graphs the problems of deciding monopolarity and polarity can be solved in polynomial time, and they gave $O(n + m^3)$ and $O(n + m^4)$ time algorithms for these problems, respectively. Later, in [44] the same authors improved the above algorithms giving $O(nm)$ and $O(nm^2)$ algorithms for monopolar and polar recognition of permutation graphs, respectively.

As Ekim et al. mention for the first time in [43], since "admitting a polar partition" can be expressed in monadic second order logic without using edge set quantification, polar graphs of bounded treewidth or bounded clique-width can be recognized in polynomial time by the results of [3, 35] and [36]. Cographs, for example, are the graphs which clique-width is at most 2 [37], then the polynomial time recognizability of polarity on cographs is proved independently of the results in [47][5]. However, chordal graphs and permutation graphs have unbounded treewidth and clique-width [69], which makes interesting the results in [45] and [43].

In [78] Huang and Xu defined a line-polar graph as a graph which its line-graph is a polar graph. In the same paper, they proved a forbidden subgraph characterization of line-polar bipartite graphs by several infinite families of minimal non line-polar bipartite graphs, answering a question raised in [47]. The results in [78] were useful for Ekim and Huang to give an $O(n)$ time algorithm which decides whether the line-graph of an input bipartite graph of order n is polar, and constructs a polar partition when one exists [46].

[5]Other classes with bounded treewidth or clique-width are distance-hereditary graphs and outerplanar graphs

Nevertheless, shortly after the last mentioned result was published, and making use of the basic ideas used in it, an $O(n)$ algorithm for recognizing general line-polar graphs was given by Churchley and Huang [27].

Talmaciu and Nechita characterized the graphs that are both trivially perfect and polar [103]. They gave an $O(n(n+m))$ algorithm for recognizing these graphs, which in turn can be used to calculate in the same time the independence and clique number of the input graph. Furthermore, they observed that every connected polar trivially perfect graph has domination number 1.

Churchley and Huang proved in [29] that recognizing polarity or monopolarity remains NP-complete for triangle-free graphs. Thus, since the complements of triangle-free graphs are claw-free graphs and polarity is closed under taking complements, they concluded that recognizing polarity also remains NP-complete for claw-free graphs. In addition, they proved in the same paper, based on an structural characterization of monopolar claw-free graphs, that monopolarity can be recognized by means of an $O(n^3)$ time algorithm that obtain a monopolar partition if one exists. It is worth mentioning that in [28] the same authors of [29] exhibited an $O(n^2m)$ time algorithm for recognizing monopolar claw-free graphs in its list version[6]. The results on claw-free graphs given by Churchley and Huang are particularly interesting since in [27] was shown that both polarity and monopolarity can be recognized in linear time for line graphs, and it is well known that line graphs are claw-free graphs. Taking as inspiration the basic ideas of [29], Churchley and Huang also unified in [30] the results on polynomial time monopolarity recognition for cographs [47], chordal graphs [45], line graphs [27], claw free graphs [29] and permutation graphs [43]. They described a class of graphs $\mathcal{G}$ including all the classes alluded above, and proved for the graphs in $\mathcal{G}$ the existence of an $O(n^4)$ time algorithm to determine if they are monopolar and find a monopolar partition if one exists. Graphs which contains no induced cycle of length k, with $k \geq 5$, are contained in the class $\mathcal{G}$, particularly, the complements of comparability graphs are in such a class. The latter observation answered the question formulated in [43] about whether the co-comparability polar graphs can be recognized in polynomial time.[7]

[6]In several papers is mentioned the manuscript *"Polarity of claw-free graphs"* of Tinaz Ekim where, apparently, an $O(n^4m^2)$ algorithm for recognizing monopolar claw-free graphs was given [27, 29, 30, 46]. Nonetheless, this result could not be corroborated.

[7]The authors of [30] mentioned that they were able to show that recognizing monopolar comparability graphs is NP-complete, but they cite a manuscript, *"Remarks on monopolarity"*, that apparently was never published.

In [30] was also exhibited an alternate $O(n^2m)$ time algorithm to the given in [105] for determining whether a given graph is unipolar. The authors finished their article proving that if it is known that a graph G admits a polar partition $\{A, B\}$, then it can be decided in $O(n^2m)$ time whether G is monopolar, which is interesting since, as we have said above, monopolar graph recognition is NP-complete in general [54].

In [83] and [84] Le and Nevries focused on the problem of recognizing polar and monopolar graphs that are also planar. Their main results included in their papers are summarized below.

First, they showed that recognizing both, polarity and monopolarity, remains NP-complete either on triangle-free planar graphs of maximum degree 3 or on $\{C_4, \ldots, C_k\}$-free planar graphs of maximum degree 3, where k is a fixed integer greater or equal than 4. As an interesting consequence of the above results and Brooks' Theorem they concluded that recognizing polarity or monopolarity on 3-colorable graphs is NP-complete. Besides, they also proved that recognizing polarity remains NP-complete on both claw-free co-planar graphs and $\{2K_2, C_5\}$-free graphs, and therefore on hole-free graphs[8] and P_5-free graphs.

The previous results deserve the following contrasts. Since the graphs of maximum degree 2 are disjoint unions of paths and chordless cycles, the graphs on this class are always monopolar, so the results are best possible respect to maximum degree. Furthermore, as we have already mentioned, chordal graphs (i.e. $\{C_4, C_5, \ldots\}$-free graphs) admit polynomial time polarity and monopolarity recognition algorithms, and then the result on $\{C_4, \ldots, C_k\}$-free planar graphs is also best possible in terms of forbidden induced cycles. In addition, a graph is bipartite if and only if it is 2-colorable, and we have observed that every bipartite graph is monopolar and hence polar, thus, recognition of monopolarity and polarity are trivial on that family of graphs, but for the class of 3-colorable graphs both problems are NP-complete.

Besides of their results on NP-completeness of polar and monopolar recognition of graph classes, Le and Nevries also exhibited some graph classes on which these problems can be solved efficiently[9]. The outstanding results given on monopolarity recognition includes polynomial time algorithms for 2-connected hole-free graphs and maximal planar graphs, as well as polynomial time al-

[8]Hole-free graphs are the $\{C_5, C_6, \ldots\}$-free graphs. Note that every chordal graph is a hole-free graph.

[9]Actually, the authors worked with a more general version of monopolarity, which they called monopolar extension.

gorithms on a superclass of chair-free graphs and a superclass of hole-free graphs. Some relevant graph classes included in these superclasses are claw-free graphs, line graphs, cographs, permutation graphs, chordal graphs, P_5-free graphs (hence $2K_2$-free graphs) and locally maximal planar graphs (i.e. graphs whose blocks are maximal planar), so these results generalize several of the previous known.

As polarity recognition remains NP-complete for planar graphs, the authors of [83] also included in their research the identification of some planar graphs subclasses for which polarity is polynomial time decidable. Only one of these graph classes was known before their results, outerplanar graphs, for which polarity recognition can be solved in polynomial time since they have bounded treewidth. As a corollary of a more general result, they showed that polarity recognition is possible in polynomial time on hole-free planar graphs and chair-free planar graphs, as well as on every hereditary subclass of planar graphs such that *monopolar extension* can be solved in polynomial time, as happens with maximal planar graphs.

An interesting remark of the work of Le and Nevries is that they showed that polarity recognition is NP-complete on co-planar graphs, $2K_2$- free graphs, P_5-free graphs, chair-free graphs and hole-free graphs, but that monopolarity is tractable on the same graph classes. In this sense, their results go nicely with those of Churchley and Huang [29], who proved that also for claw-free graphs polarity recognition remains hard while monopolarity recognition has an efficient algorithm.

As we have pointed above, in [83] and [84] was proved that recognizing polarity and monopolarity on 3-colorable graphs is NP-complete. Additionally, Churchley and Huang announced in [30] an unpublished work proving that polarity of comparability graphs is also NP-complete. In relation with these works, but independently to them, Yolov showed in [107] that monopolarity and polarity remains NP-complete restricted to 3-colorable comparability graphs. In [83] was formulated the question whether there exists a class for which polarity is easier than monopolarity. About this question, Yolov proved in his paper that polarity is not easier than monopolarity for graph classes closed under disjoint union.

The class of generalized split graphs is the union of the unipolar and co-unipolar graphs classes. Research on recognizing generalized split graphs was done on [52] and [91], where $O(n^3)$ and $O(n^2)$ time algorithms with such a purpose were found, respectively.

Table 2.1 summarizes the complexity status of polarity and monopolarity of some relevant graph classes.

Graph class	**Monopolarity**	**Polarity**
general graphs	NP-c [54]	NP-c [23]
chordal	$O(n+m)$ [45]	P [45]
line graphs	$O(n)$ [27]	$O(n)$ [27]
permutation	$O(nm)$ [44]	$O(nm^2)$ [44]
maximal planar	$O(n^4)$ [84]	P [84]
3-colorable	NP-c [84]	NP-c [84]
comparability	NP-c [107]	NP-c [107]
co-comparability	P [30]	NP-c [107]
outerplanar	?	P [3, 35]
trivially perfect	?	$O(n(n+m))$ [103]
distance-hereditary	?	P [36]
triangle-free	NP-c [29]	NP-c [29]
cographs (P_4-free)	$O(n)$ [47]	$O(n)$ [47]
claw-free	$O(n^3)$ [29]	NP-c [29, 84]
P_5-free	$O(n^4)$ [84]	NP-c [84]
chair-free	$O(n^4)$ [84]	NP-c [28, 84]
$\{2K_2, C_5\}$-free	$O(n^4)$ [84]	NP-c [84]
hole-free	P [84]	NP-c [84]
$\{C_4, C_5\}$-free planar	NP-c [84]	NP-c [84]
triangle-free planar	NP-c [28, 84]	NP-c [28, 84]
claw-free planar	$O(n^3)$ [29]	P [84]
claw-free co-planar	$O(n^3)$ [28]	NP-c [84]
chair-free planar	P [83]	P [83]
hole-free planar	P [83]	P [83]
3-colorable comparability	NP-c [107]	NP-c [107]
comparability co-comparability	P [43]	P [43]

Table 2.1: P, NP-c, and ? means that the complexity status of the corresponding problem is polynomial, NP-complete or unknown, respectively.

2.3.1 (s,k)-polarity and matrix partitions

Recall that a graph G is polar if $V(G)$ admits a polar partition $\{A, B\}$ such that A induces a complete multipartite graph and B induces a cluster. For non

negative integers s and k, we will say that G is (s,k)-***polar*** if $V(G)$ admits an (s,k)-***polar partition***, that is, a polar partition $\{A,B\}$ such that $G[A]$ has at most s parts and $G[B]$ has at most k connected components.

Observe that the complement of an (s,k)-polar graph is an (k,s)-polar graph, and that if a given graph is (s,k)-polar, then it is (t,ℓ)-polar for every pair of non negative integers t and ℓ such that $s \leq t$ and $k \leq \ell$. Also notice that monopolar graphs, before defined as the graphs that admit a polar partition $\{A,B\}$ where A is an independent set, are exactly the (s,k)-polar graphs such that $s = 1$. An example of a $(2,2)$-polar graph and a $(2,2)$-polar partition are shown on the Figure 2.6.

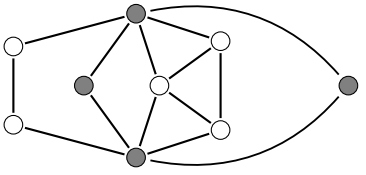


Figure 2.6: The graph shown is a $(2,2)$-polar graph since the shaded vertices induce a complete bipartite graph and the non-shaded vertices induce the disjoint union of two complete graphs.

As we mentioned before, matrix partitions are a generalization of many interesting problems of partitions in graphs, and polarity is not the exception. In fact, it is easy to see that an (s,k)-polar partition of a graph G corresponds to an M-partition in which the matrix M is such that the A block is an $s \times s$ matrix obtained from an identity matrix by exchanging 0's and 1's, the B block is the $k \times k$ identity matrix, and the C block is a $s \times k$ matrix with only $*$ entries. It is worth noting that, as a case of the more general results on matrix partitions presented in [56], for any fixed s and k, the class of (s,k)-polar graphs can be recognized in polynomial time; unfortunately, a detailed explanation of this fact goes beyond from the scope of this text.

For any fixed non negative integers s and k, having an (s,k)-polar partition is a hereditary property, and thus, (s,k)-polar graphs can be characterized by a (possibly infinite) family of forbidden induced subgraphs. A stronger result was proved by Feder, Hell and Xie in [57]; they showed, in the context of matrix partitions, that if M is a matrix with A and B blocks of size $k \times k$ and $\ell \times \ell$, respectively, that has no $*$ entries in A and in B, and such that or C

has only $*$ entries, or C has no $*$ entries, then M-partitionable graphs can be characterized by a finite set of forbidden subgraphs; as we have observed, such is the case of (s,k)-polar graphs, so it follows that for any fixed non negative integers s and k, (s,k)-polar graphs can be characterized by a finite family of forbidden induced subgraphs.

Despite the aforementioned, only for few very small values of s and k the family of minimal (s,k)-polar obstructions is known; a graph is $(0,k)$-polar if and only if it is a disjoint union of at most k cliques (or equivalently, if it is $\{P_3, \overline{K_{k+1}}\}$-free), it is $(s,0)$-polar if and only if it is a complete s-partite graph (i.e., if it is $\{\overline{P_3}, K_{s+1}\}$-free), and it is $(1,1)$-polar if and only if it is a split graph (and thus, by the results in [58], if and only if it is $\{2K_2, C_4, C_5\}$-free).

Chapter 3

Cographs

As Corneil, Lerchs and Stewart Burlingham pointed in [33], the graphs that are currently called *cographs* were independently discovered in the early 1970s by various researchers, and they were introduced through very different approaches. In consequence, cographs had been named in various ways, like hereditary Dacey graphs [102], D^*-graphs [80] or 2-parity graphs [19]. Corneil et. al. unified in their paper all these graph classes that seemed to be different under the name of *complement reducible graphs*, or ***cographs*** for short, and they gave their own recursive definition for this class of graphs by the three following rules:

1. A trivial graph is a cograph.
2. The complement of a cograph is a cograph.
3. The disjoint union of two cographs is a cograph.

Related previous work to [33] is diverse, but it has in common that it involves graph classes which are equivalent to P_4-free graphs. Some examples of the aforementioned are the following: Seinsche showed in a paper related to colorability that a graph G is P_4-free if and only if for every non-trivial induced subgraph H of G, H or $\overline{H}$ is disconnected [98]; Sumner proved that a graph is an hereditary Dacey graph (graphs that arose in the context of empirical logic) if and only if it does not contain P_4 as induced subgraph [102]; Jung proved in [80] that P_4-free graphs (which he named D^*-graphs) are exactly the comparability graphs of a special kind of posets that he named *multitrees*; Burlet and Uhry showed that their *2-parity graphs*, that are the graphs such that the diameter of each of its connected components is at most two, exactly match with the P_4-free graphs [19].

Corneil, et. al. extended the known characterizations of P_4-free graphs to a list of eight equivalences, including their own recursive definition of cographs, and after that, various new characterizations of cographs have been given. In the following section, we will introduce some brief definitions that allow understanding for some of the characterizations of cographs, and we will summarize the most important of these equivalences in a marvelous theorem.

3.1 Cograph characterizations

A graph $G = (V, E)$ is a ***Dacey graph*** if, for every maximal clique C of G and every pair of distinct vertices $u, v \in V$, we have that $uv \in E$ whenever $C \subseteq N(u) \cup N(v)$. A graph is an ***hereditary Dacey graph*** if every induced subgraph is a Dacey graph. Dacey graphs were introduced in 1970 in the context of empirical logic by Foulis [59] who proved, using other words, that a graph is a hereditary Dacey graph if and only if it is P_4-free. This result was not presented in graph terms until 1974 by Sumner [102].

Given a partially ordered set $(V, \leq)$ and $V_1, V_2 \subseteq V$, we write $V_1 \leq V_2$ if $v_1 \leq v_2$ for all $v_1 \in V_1, v_2 \in V_2$. We also denote $S(u) = \{v \in V \mid u \leq v\}$. A partially ordered set $(V, \leq)$ is called a ***series-parallel partial order***[1] if, for all $u, v \in V$, $u \leq v$ or $S(u) \setminus S(v) \leq S(u) \cap S(v)$. Jung showed in [80] that a graph has no P_4 as induced subgraph if and only if it is the comparability graph of a series-parallel partial order. Independently, Möhring proved in [93], using an equivalent definition for series-parallel partial orders, that a graph is a cograph if and only if it is the comparability graph of a series parallel partial order.

The ***clique-width*** of a graph G is the minimum number of labels needed to construct G by means of the following 4 operations:

1. Creation of a new vertex v with label i, denoted $i(v)$.
2. Disjoint union of two labeled graphs G and H, denoted $G + H$.[2]
3. Joining by an edge every vertex labeled i to every vertex labeled j, where $i \neq j$. It is denoted by $\eta_{i,j}$

[1] Jung called a poset satisfying this property a *multitree* [80], but this term is actually used for other kind of partial orders, and *series-parallel partial order* is the term that is currently most accepted for this structure.

[2] In the usual notation of clique-width, the disjoint union is denoted by $\oplus$, but for congruence with this text, here it is denoted by $+$.

4. Renaming label i to label j, denoted $\rho_{i\to j}$.

For example, the path on 4 vertices, P_4, can be constructed by means of the above operations using only 3 labels in the following way:

$$\eta_{2,3}((\eta_{1,3}(3(c)+1(d)))+(\eta_{1,2}(1(a)+2(b)))).$$

A graphical representation of the construction of P_4 is shown in Figure 3.1.

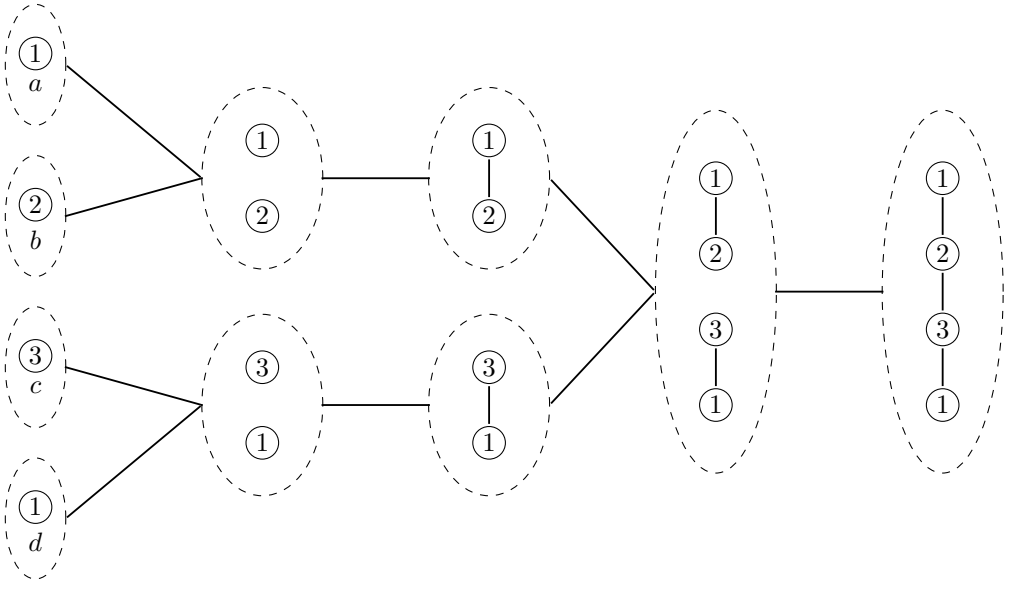


Figure 3.1: Construction of P_4 using 3 labels by means of *clique-width operations*.

In fact, the clique-width of P_4 is exactly three. Let suppose for a contradiction that P_4 can be constructed under these operations using two labels. We can assume that the last operation was not the renaming of labels, and since P_4 is connected, the last operation must have been the join of vertices with distinct labels. But we note that if we have four vertices with 2 labels and we make the join of vertices with distinct labels, we always obtain either an induced C_4 or an induced $K_{1,3}$, and no one of these graphs is a subgraph of P_4, so we have a contradiction from the assumption that P_4 can be constructed using only two labels. Thus, the clique-width of P_4 is exactly 3. A more general result is not difficult to show; cographs are exactly the graphs with clique-width at most two [37].

Recall that a permutation P of the set X is a sequence without repetitions that includes all the elements of X. A ***permutation pattern*** of the permutation P is simply a subsequence of P, and a permutation P of an initial segment of natural numbers $\{1, 2, \ldots, n\}$ is called ***separable*** if it contains neither the permutation pattern $(3, 1, 4, 2)$ nor the permutation pattern $(2, 4, 1, 3)$. Bose, Buss and Lubiw observed in [9] that a graph is P_4-free if and only if it is the permutation graph associated with a separable permutation.

Bodlaender and Möhring showed in [7] that every minimal chordal completion of a cograph is a cograph. Parra and Scheffler enlarged the previous result by showing that a graph G is P_4-free if and only if every minimal chordal completion of G is a trivially perfect graph [96]. Note that trivially perfect graphs are exactly the $\{P_4, C_4\}$-free graphs [68], and then, it follows that trivially perfect graphs are cographs that are chordal graphs. Moreover, by the result of Parra and Scheffler, trivially perfect graphs are the cographs that are chordal graphs. Many characterizations of trivially perfect graphs were proved in [106], including that they are the cographs that are interval graphs.

As we have observed in Section 2.1, for any given graph, its Grundy number is greater than or equal to its chromatic number, and the latter always is at least as big as its clique number. In symbols, for any graph G,

$$\omega(G) \leq \chi(G) \leq \Gamma(G).$$

A perfect graph is, as we defined above, a graph G such that, for every induced subgraph H of G, $\omega(H) = \chi(H)$. Christen and Selkow extended the concept of perfect graph in the following way: for distinct members, α and β, of the set $\{\omega, \chi, \Gamma\}$, a graph G is called an $\alpha\beta$***-perfect graph*** if $\alpha(H) = \beta(H)$ for every induced subgraph H of G [24]. Thus the perfect graphs simply correspond to the $\omega\chi$-perfect graphs.[3]

Evidently, every $\omega\Gamma$-perfect graph is a $\chi\Gamma$-perfect graph. Christen and Selkow not only proved that these graph classes are exactly the same but that furthermore, they coincide with the graphs that do not contain an induced subgraph isomorphic to P_4. As a corollary, the authors also proved that a graph G is $\chi\Gamma$-perfect if and only if, for each induced subgraph H of G,

$$\Gamma(H) + \Gamma(\overline{H}) \leq |V(H)| + 1.$$

[3]The graphs such that its Grundy number equal its chromatic number are currently called ***well-colored graphs***. A ***hereditary well-colored graph*** is a graph such that every of its induced subgraphs is a well-colored graph, i.e., a hereditary well-colored graph is a $\chi\Gamma$-perfect graph.

In addition, based on the results of Seinsche [98], Christen and Selkow demonstrated that a graph is $\chi\Gamma$-perfect if and only if it is a cograph.

A digraph is ***transitive series parallel***, or *TSP*, if it is transitive and does not contain an induced subdigraph isomorphic to the digraph shown in Figure 3.2 Corneil, Lerchs and Burlingham showed in [33] that a graph G is a cograph if and only if there exist an orientation D of G such that D is a TSP digraph.

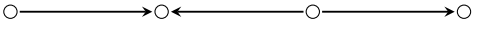


Figure 3.2: The only obstruction for TSP digraphs.

After the summary of concepts necessary to give the diverse known characterizations of cographs, we finally can present a great theorem that relates them. Most of the equivalences of Theorem 3.1 have been already justified along this chapter, and for a complete warranting of it, it is only necessary to mention a few other results.

As we have mentioned before, Chvátal proved that a linear vertex ordering $<$ of G is perfect if and only if there is no induced P_4, (a, b, c, d), such that $a < b$ and $d < c$ [31]. It is a simple consequence of the latter equivalence that a graph G is P_4-free if and only if every linear vertex order of G is a perfect order.

Seinsche showed that a graph G is P_4-free if and only if the complement of any nontrivial connected induced subgraph of G is disconnected [98]. The author also mentioned (without a proof) that the P_4-free graphs are the graphs that can be obtained from K_1 by means of vertex doublings. However, it is a simple exercise to prove that a graph obtained from K_1 by means of successive vertex doublings is a P_4-free graph, and the proof of the converse can be found in [19]. The equivalence of the sentences $1, 3, 4, 5$ and 6 was proved by Corneil, Lerchs and Burlingham using the results of Seinsche [33].

Cographs also admit a characterization as the graphs that can be generated from single vertex graphs by means of disjoint union and join operations. This result was explicitly mentioned by first time in [38] by Damaschke, who said that the proof of the equivalence could be found in [33], and although a direct proof of the fact is not in that paper, it can be easily obtained from the results in it. It is easy to note that every graph that satisfies the sentence 6 of Theorem 3.1 can be constructed from single vertex graphs by means of disjoint union and join operations (see the construction of *cotrees* in the following section).

Furthermore, since the graph join $G \oplus H$ is just the graph $\overline{(\overline{G} + \overline{H})}$, it follows that every graph obtained from copies of K_1 by using disjoint union and join operations is a cograph. The above observations together with the proof of the equivalence of sentences 1 and 6 of Theorem 3.1 given in [33] implies the characterization of cographs mentioned by Damaschke. For a summary of a possible proof sketch of Theorem 3.1 we provide the scheme in Figure 3.3, after the statement of the theorem.

Theorem 3.1. *Let G be a graph. Then the following sentences are equivalent.*

1. *G is a cograph.*
2. *G can be obtained from single vertex graphs by means of join and disjoint union operations.*
3. *Any nontrivial induced subgraph of G has a pair of (true or false) twins.*[4]
4. *For every induced subgraph H of G, every maximal clique of H has exactly one vertex in common with every maximal independent set of H.*[5]
5. *G is a P_4-free graph.*
6. *The complement of any nontrivial connected induced subgraph of G is disconnected.*
7. *G is an hereditary Dacey graph.*
8. *Every connected induced subgraph of G has diameter at most two.*
9. *G is the comparability graph of a series-parallel partial order.*
10. *G is a graph with clique-with at most 2.*
11. *G is a permutation graph of a separable permutation.*
12. *Every chordal completion of G is a trivially perfect graph.*
13. *Every vertex order of G is a perfect order.*
14. *There exists an orientation D of G such that D is a TSP digraph.*

[4] An equivalent way to express this property is the following: G can be obtained from K_1 by repeat doubling of one vertex with or without joining the two doubles.

[5] In [33] this property of H is called the clique-kernel property, or CK-property.

15. G is a $\omega\Gamma$-perfect graph.

16. G is a $\chi\Gamma$-perfect graph (i.e., G is a hereditarily well-colored graph).

17. For every induced subgraph H of G, $\Gamma(H) + \Gamma(\overline{H}) \leq |V(H)| + 1$.

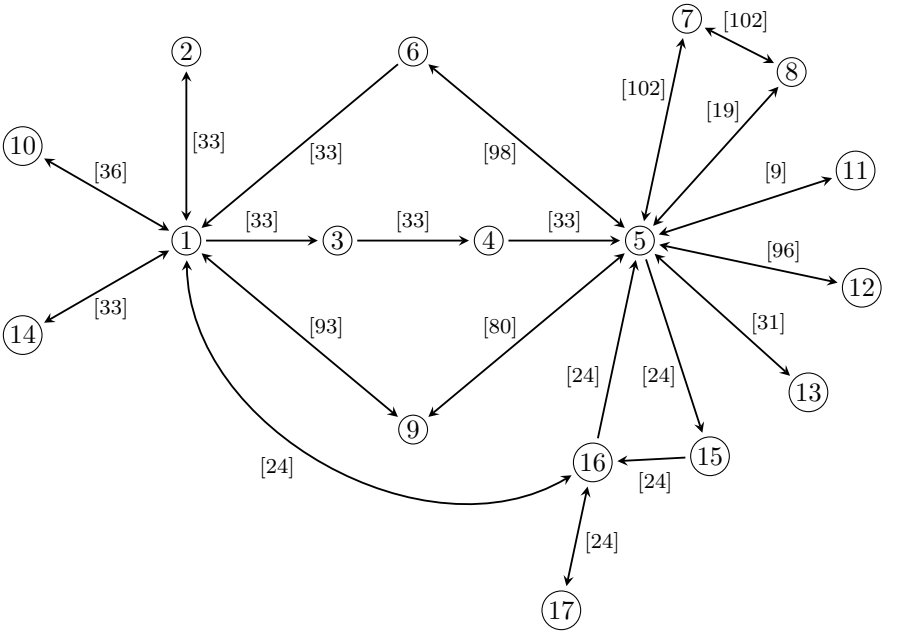


Figure 3.3: A possible proof sketch of Theorem 3.1. Each vertex correspond to a sentence of the theorem and the label of each arc correspond to a paper where the proof of the corresponding implication can be found.

Although many interesting properties of cographs follow by Theorem 3.1, we want to make emphasis on two of them that will be particularly important for the development of our results in Chapter 4: since cographs are the P_4-free graphs, *to be a cograph* is a hereditary property, and the complement of any cograph is also a cograph.

3.2 Structural decomposition of cographs

In [33] Corneil, Lerchs and Burlingham showed that every cograph can be uniquely represented by a tree such that their leaves correspond to the vertices

of the cograph. In this section, we will introduce that representation, and in the following, we will show how it has been used to derive some structural and algorithmic properties of cographs.

3.2.1 Cotrees

Given a tree T, a vertex of T is called an ***inner vertex*** if its degree is at least two, and it is called a ***leaf*** if it is a vertex of degree one. A ***rooted tree*** is an ordered pair (T, r) where T is a tree and r is a vertex of T that has been designated as the ***root***; a rooted tree (T, r) will be simply referred by T when the specification of its root is not necessary. We will denote the only path from a leaf x to the root of a given rooted tree T as $P_T(x)$, and we establish a partial ordering on the vertex set of T by the relation $u \geq v$ if and only if u is a vertex of $P_T(v)$; if distinct vertices u and v of T are such that $u \geq v$ we will say that u is an ***ancestor*** of v, and that v is a ***descendant*** of u. The ***lowest common ancestor*** of the vertices u and v is the only vertex which is in both of $P_T(u)$ and $P_T(v)$ at the furthest distance from the root.

For a given graph G, any function f from a subset of $V(G)$ into a set L will be called an ***L-labeling function*** for G, and we will say that G is a graph ***L-labeled*** (by f).

According to the Theorem 3.1, a graph G is a cograph if and only if for every nontrivial induced subgraph H of G, H or $\overline{H}$ is disconnected. The above characterization not only justifies the name of *complement reducible graphs*, but it also provides of a unique tree representation for every cograph.

Given a cograph G, we can obtain a $\{0, 1\}$-labeled rooted tree T containing the whole structure of G by means of the following construction. The root of T is $V(G)$, and for every vertex A of T such that $|A| \geq 2$, if $G[A]$ is disconnected (respectively connected), then the children of A are all the different subsets of A inducing a component of $G[A]$ (respectively $\overline{G[A]}$). For each inner vertex A of T, if $G[A]$ is disconnected, A is assigned the label 0, otherwise A is assigned the label 1; the leaves of T will be not labeled. The rooted tree constructed above is the ***cotree*** associated to the cograph G. Note that if we apply this procedure of cotree construction to a general graph G, it will finish if and only if G is a cograph, and it will fail (entering in an infinite loop) if and only if there exists a subset A of $V(G)$ with $|A| \geq 2$ such that both of $G[A]$ and $\overline{G[A]}$ are connected.

Evidently, the cotree for a particular cograph is unique, and it is usually drawn with the root at the top, the inner vertices labeled with 0 or 1, and

the leaves of the tree, that are singletons of the cograph, identified with the only vertex that they contain. An example of a cograph and the step by step construction of its cotree are shown in Figure 3.4.

It is worth noting that there are other very similar notions of "cotree". Our definition coincides with the one described in [93] and [7], but it differs slightly with the original one given in [33].

The next theorem follows easily from the cotree definition.

Theorem 3.2. *Let G be a cograph and let T its associated cotree.*

1. *The cotree $\overline{T}$ of $\overline{G}$ is exactly T with 0 and 1 labels interchanged.*

2. *An induced rooted subtree of T is the cotree of the corresponding induced subgraph of G.*

3. *For any leaf x of T, the vertices of the unique path from x to the root of T have alternating labels 0 and 1, except for x, that has no label assigned.*

4. *The vertices u and v of G are adjacent if and only if the lowest common ancestor of u and v in T is labeled with 1.*

5. *Each labeled vertex of T has at least two children.*[6]

3.3 Computational properties of cographs

Various algorithmic problems that are difficult for general graphs may be solved in polynomial time for cographs, in fact, the cotrees are the basis for many efficient algorithms. Nevertheless, as was pointed out in [34], "before such algorithms can be employed it is necessary to have a fast cograph recognition algorithm which also produces the cotree recognition of a cograph".

At following, we will make first a brief summary of some algorithms for cograph recognition, and then, we will mention some outstanding algorithmic problems that are remarkable when restricted to cographs.

[6]For the original definition of cotree, given in [33], this property is changed for the following: "Every labeled vertex, except possibly the root, will have two or more children; the root will have only one child when the represented cograph is disconnected". All other properties of the theorem also follows with the original definition of cotree.

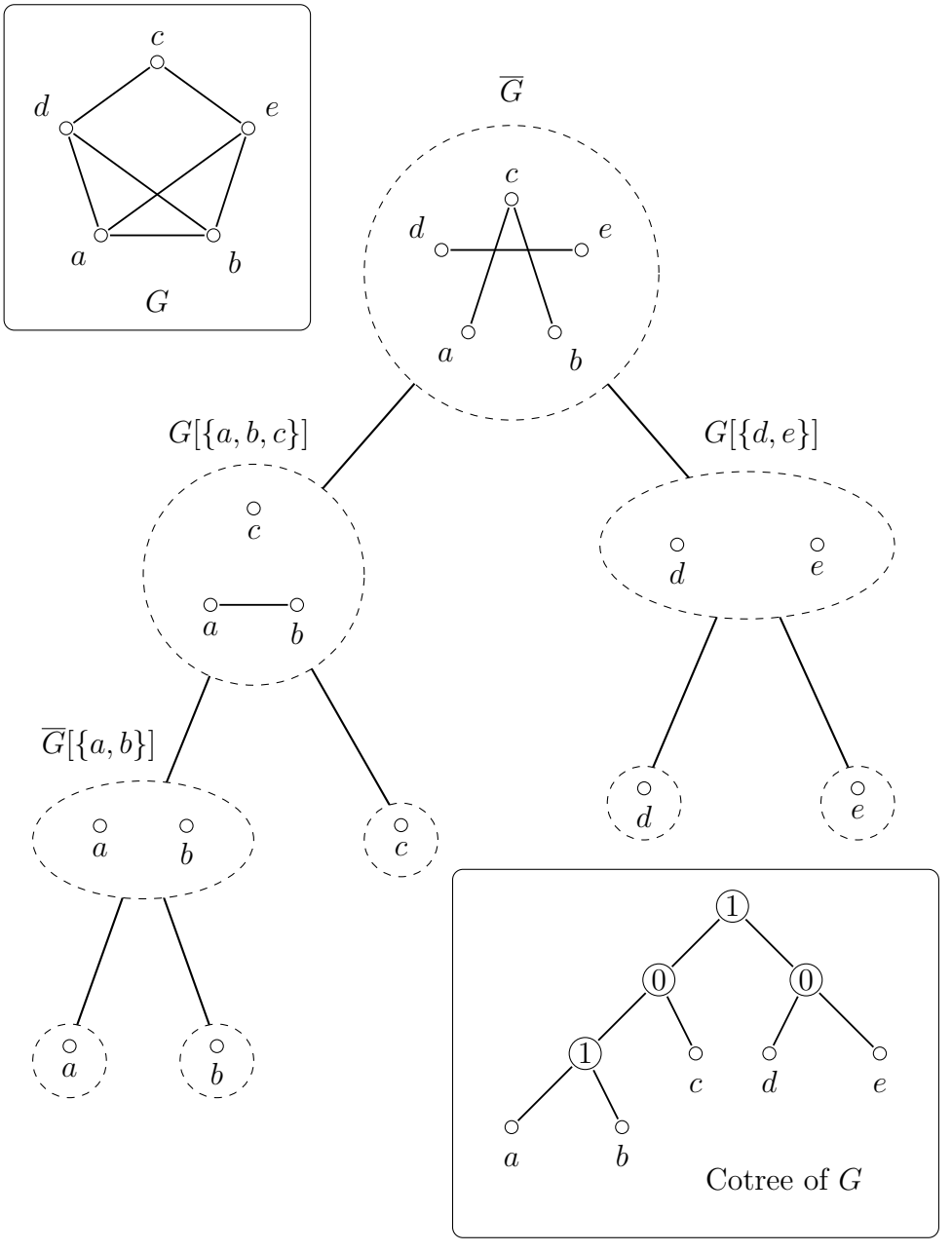


Figure 3.4: Detailed construction of the cotree associated to the cograph G.

3.3.1 Cograph recognition and cotree construction

Many algorithms for cograph recognition have been developed since the introduction of these graphs; the intent of this section is not to give a complete survey of them, but rather to mention some of these algorithms that are remarkable in a historical or practical sense.

It follows from the characterizations given for cographs in Theorem 3.1 that these graphs can be recognized in polynomial time; for example, since cographs are exactly the P_4-free graphs, to verify if a graph G is a cograph, we can simply check whether each induced subgraph of G with 4 vertices is isomorphic to P_4. Nevertheless, this algorithm is not very efficient, and although it provides a proof if G is not a cograph by exhibiting a P_4, it does not give a certificate in the positive case.

Corneil, Perl and Stewart presented the first algorithm with complexity $O(n + m)$ for recognition of cographs and construction of its corresponding cotree [34]. As is described by its authors, this algorithm is *incremental*, in the sense that the vertices are processed one at time until the entire graph has been handled; they base on the fact that cographs are hereditary, and having as input a cograph G with cotree T, the algorithm determines if the graph obtained by the adding of a new vertex to G is also a cograph and, if so, modifies T to be the cotree of the new cograph. Unfortunately, this algorithm does not return a certificate in the negative case.

From the above algorithms, a natural expectation for a cograph-recognition algorithm would be to construct the associated cotree in the positive case and to exhibit a forbidden substructure in the negative case. An algorithm as above with complexity $O(n^3)$ that is remarkable for its simplicity was exhibited by Möring in 1989 [93]. This algorithm is the base of the cotree construction we exposed in Section 3.2; it returns the corresponding cotree if the input graph is a cograph, and otherwise, it returns a nontrivial connected induced subgraph H of G with connected complement $\overline{H}$, which is a proof that G is not a cograph by sentence 6 of Theorem 3.1.

A much faster algorithm for cograph recognition that certifies positive an negative answers was achieved by Capelle and Cournier in [20] by completing the incremental algorithm given by Corneil, Perl and Stewart and adapt it in order to produce an induced P_4 with complexity $O(|N_G(v)|)$ in case of a fail while introducing a new vertex v in the cotree.

Another $O(n + m)$ complexity algorithm for cograph recognition was given recently in [15] by Bretscher, Corneil, Habib and Paul; it returns the associated

cotree if the input graph is a cograph and returns an induced P_4 otherwise. This algorithm is a simpler algorithm than the one given in [34], and due to it is based on LexBFS, it posses a straightforward implementation. By its linear complexity, easy implementation and simplicity, this is one of the best cograph-recognition algorithms existing to the date.

3.3.2 Easy and difficult problems on cographs

Since cotree representation of cographs is unique, linear time tree isomorphism algorithms, as the given in [76], can be applied to cotrees for obtaining linear algorithms that solve the COGRAPH ISOMORPHISM PROBLEM (the problem of determining whether two cographs are isomorphic). In contrast, both of the SUBGRAPH ISOMORPHISM PROBLEM and INDUCED SUBGRAPH ISOMORPHISM PROBLEM (the problems of verify if a given graph H is a subgraph, or induced subgraph, of another given graph G) remains NP-complete when restricted to cographs (see [33] and [39], respectively).

Cograph algorithms for calculating parameters typically assign weights to the leaves of the cotree, and associate operations with the 0-nodes or 1-nodes of it. Then, these algorithms use the operations on the inner nodes of the cotree to calculate particular values associated to the subgraphs induced by the rooted subtrees at the inner nodes, and at the end, the value assigned to the root of the cotree is the value assigned to the cograph. Polynomial algorithms of this type for determining diverse parameters of cographs as the chromatic number, clique number, number of transitive orientations and scattering number[7], have been given in [33]. It is worth nothing that Jung showed in [80] that the scattering number of a cograph is closely related with the existence of Hamiltonian paths and cycles, specifically, he showed the following theorem.

Theorem 3.3. *Let G be a cograph. Then*

1. *G has a Hamiltonian path if and only if $s(G) \leq 1$,*
2. *G is Hamiltonian if and only if $s(G) \leq 0$ and $|V(G)| \geq 3$,*
3. *G is Hamiltonian-connected if and only if $s(G) < 0$.*

[7]The ***scattering number*** of a graph G is the maximum number $c(G-S)-|S|$ such that $S \subseteq V$, $c(G-S)$ denotes the number of components of $G-S$, and $c(G-S) \neq 1$.

Thus, as the scattering number can be calculated in polynomial time for cographs, the result of Jung shows that Hamiltonicity can be solved efficiently for these graphs.

Cographs also inherit efficient algorithms from some superclasses to which they belong. Some outstanding results of this type are summarized in Table 3.1.

Problem	**Graph superclass**	**Complexity**
Cliquewidth decompositon	Distance hereditary	Linear [69]
Treewidth decomposition	Distance hereditary	Linear [17]
	Permutation	Linear [89]
Clique	Comparability	Linear [68]
Domination	Distance hereditary	Linear [94]
	Permutation	Linear [53]
Independent set	Permutation	Linear [95]
Monopolarity	Permutation	Polynomial [44]
Polarity	Permutation	Polynomial [44]

Table 3.1: Some problems that can be solved efficiently in some superclass of cographs.

Chapter 4

(s,k)-polar cographs

4.1 Previous results

In 1990, Peter Damaschke proved that the class of cographs is well-quasi-ordered by the induced subgraph relation[1] [38]; he concluded from this result that every hereditary property of cographs can be characterized by a finite family of forbidden induced subgraphs.

A problem that arises in a natural way from the result of Damaschke is, giving a hereditary property of graphs, to find the family of minimal forbidden induced subgraphs characterizing such a property in the class of cographs. Notice that the knowledge of such families may have two interesting applications, one of them providing light in some problems at analyzing the structure of the members of these families. Also, the minimal forbidden induced subgraphs of a cograph subclass can be used as no-certificates for the decision problem associated with the recognition of the graphs in such a subclass. For example, since (s,k)-polarity is a hereditary property, if we know the complete family of minimal forbidden induced subgraphs of the (s,k)-polar cographs, for some fixed s and k, an algorithm could be designed to receive a cograph G, decide if it has an (s,k)-polar partition, and return either the (s,k)-polar partition of G (a yes-certificate) or one of the forbidden induced subgraphs (that is, a no-certificate).

For any pair of fixed non negative integers, s and k, a cograph which is not (s,k)-polar will be called a a ***cograph (s,k)-polar obstruction***. A cograph

[1] A poset $(M, \leq)$ is called a *well-quasi-ordering* if every infinite sequence $(a_1, a_2, \dots)$ of elements of M includes a pair a_i, a_j such that $i < j$ and $a_i \leq a_j$.

which is not (s,k)-polar, but is such that every proper induced subgraph is, will be called a ***cograph minimal*** (s,k)***-polar obstruction***.

Ekim, Mahadev and de Werra considered by first time the class of polar cographs in [47]. They provided a characterization in terms of eight forbidden subgraphs for this class, and gave an algorithm in time $O(n)$ for finding a largest induced polar subgraph in cographs, which also serves as a polar cograph recognition algorithm. Additionally, they also examined the (s,k)-polar graphs such that $\min\{s,k\} \leq 1$, for which a characterization by sixteen forbidden subgraphs was also given [48].

The authors of [47] also proposed, as a natural continuation of their work, the problem of finding a characterization of $(2,2)$-polar cographs by forbidden subgraphs; this problem was recently solved by Hell, Hernández-Cruz and Linhares-Sales in [74], where they proved that there are 48 cograph minimal $(2,2)$-polar obstructions. Independently, Bravo, Nogueira, Protti and Vianna exhibited in [14] the exhaustive list of nine cograph minimal $(2,1)$-polar obstructions; by taking complements it is trivial to obtain analogous results for $(1,2)$-polar cographs. The above characterizations are all the known results on complete lists of cograph minimal (s,k)-polar obstructions, that is, these lists are known only for $0 \leq s,k \leq 2$, and for the case when $\min\{s,k\} \leq 1$.

4.2 Cograph minimal $(s,1)$-polar obstructions

Next we will focus on the problem of characterizing, by forbidden induced subgraphs, the $(s,1)$-polar cographs for an arbitrary fixed non negative integer s. In the rest of this chapter, we develop a recursive complete characterization of the forbidden induced subgraphs for $(s,1)$-polar cographs. We also show that cographs having an $(s,1)$-polar partition for some arbitrary non-fixed integer $s \geq 2$, can be characterized by a family of four forbidden induced subgraphs.

4.2.1 Disconnected obstructions

Observe that if a graph G admits an $(s,1)$-polar partition $\{A,B\}$, then B is just a clique, and $G[A] = G[V \setminus B]$, is a complete multipartite graph, or equivalently, a $\overline{P_3}$-free graph. On the other hand, if G is a graph containing a clique K such that $G[V \setminus K]$ is a complete multipartite graph, then clearly $\{V \setminus K, K\}$ is an $(s,1)$-polar partition of G for some integer s. These simple observations are contained in the following remark.

Remark 4.1. Let G be a cograph. If for every clique K of G, the induced subgraph $G[V \setminus K]$ contains $\overline{P_3}$ as an induced subgraph, then G is not an $(s, 1)$-polar cograph for any positive integer s.

The next lemma identifies some graphs that are cograph minimal $(s, 1)$-polar obstructions for every integer s, with $s \geq 2$. We name such graphs ***essential obstructions***.

Lemma 4.2. *The graphs $K_1 + 2K_2, \overline{K_2} + C_4, 2P_3$ and $K_1 + (\overline{P_3} \oplus \overline{K_2})$ depicted in Figure 4.1 are cograph minimal $(s, 1)$-polar obstructions for every integer s, $s \geq 2$.*

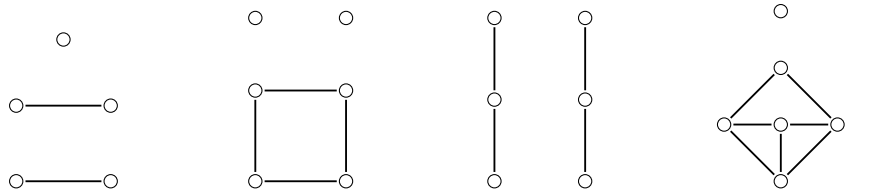


Figure 4.1: Essential obstructions.

Proof. Since all graphs shown in Figure 4.1 are P_4-free, all of them are cographs, and by a simple exploration taking into consideration Remark 4.1, it is routine to verify that none of these graphs is an $(s, 1)$-polar cograph for any positive integer s. Furthermore, it is easy to verify that in each of these graphs the deletion of any vertex results in a $(2, 1)$-polar cograph, and therefore, in an $(s, 1)$-polar cograph for every integer s, $s \geq 2$. Then, all graphs depicted in Figure 4.1 are cograph minimal $(s, 1)$-polar obstructions for any integer s greater than or equal to 2. □

Since all essential obstructions are small disconnected graphs, one can intuit that they will prevent the existence of larger disconnected minimal obstructions. Our next lemma corroborates this intuitive idea by showing that disconnected cograph minimal $(s, 1)$-polar obstructions have at most two components, except for $K_1 + 2K_2$ and $\overline{K_2} + C_4$; additionally, some restrictions on the structure of such minimal obstructions are also given.

Lemma 4.3. *Let s be an integer, $s \geq 2$. Then every cograph minimal $(s,1)$-polar obstruction different from K_1+2K_2 and $\overline{K_2}+C_4$ has at most two connected components.*

Moreover, if a cograph minimal $(s,1)$-polar obstruction has two connected components and it is neither $2P_3$ nor $2K_{s+1}$, then one of its components is K_1 or K_2, and its other component is not a complete graph.

Proof. Let G be a cograph minimal $(s,1)$-polar obstruction with at least three connected components. Observe that since G is not a split graph, G contains C_4 or $2K_2$ as an induced subgraph. In the former case, since G has at least three connected components, G contains $\overline{K_2}+C_4$ as an induced subgraph. For the latter case, again, noting that G has at least three components leads to conclude that G contains K_1+2K_2 as an induced subgraph. By the minimality of G, the previous observations imply that G is isomorphic to $\overline{K_2}+C_4$ or K_1+2K_2. So we have that every cograph minimal $(s,1)$-polar obstruction isomorphic to neither K_1+2K_2 nor $\overline{K_2}+C_4$ has at most two connected components.

Notice that $H=2K_{s+1}$ is a cograph $(s,1)$-polar obstruction, because every clique K of H is contained in the vertex set of a connected component of H, and therefore $H-K$ contains K_{s+1} as induced subgraph, which implies that $H-K$ is not a complete s-partite graph. Additionally, for every vertex $v \in V(H)$, $H-v$ is isomorphic to K_s+K_{s+1}, which is clearly an $(s,1)$-polar cograph, so H is a cograph minimal $(s,1)$-polar obstruction.

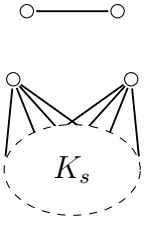
Now, suppose that G is a cograph minimal $(s,1)$-polar obstruction isomorphic to neither $2P_3$ nor $2K_{s+1}$, and with two connected components. Note that since G is $2P_3$-free, at least one of the components of G is a complete graph. If both components of G are complete graphs, then both of them must have at least $s+1$ vertices, otherwise G would be $(s,1)$-polar; but in this case G should be isomorphic to $2K_{s+1}$. Thus, we may assume that one component of G is a complete graph and the other one is not.

Finally, suppose for a contradiction that the complete component of G has three or more vertices, and let v be one of these vertices. By the minimality of G we have that $G-v$ admits an $(s,1)$-polar partition $\{A,B\}$. If B is contained in the complete component of $G-v$, then $\{A, B\cup\{v\}\}$ is an $(s,1)$-polar partition of G, a contradiction. Hence, B is contained in the non-complete component of G. Clearly, $(G-v)-B$ contains $\overline{P_3}$ as an induced subgraph, and thus, it cannot be covered by A, contradicting the choice of $\{A,B\}$ as an $(s,1)$-polar partition of $G-v$. Then the complete component of G has at most two vertices. □

In the context of matrix partitions, Feder, Hell and Hochstättler showed that if M is a matrix such that all the off-diagonal entries of the A block are equal to a, all the off-diagonal entries of the B block are equal to b, and all the entries of C are equal to c, where $a, b, c \in \{0, 1, *\}$, and the sizes of A and B are $k \times k$ and $\ell \times \ell$, respectively, then every cograph minimal M-obstruction has at most $(k+1)(\ell+1)$ vertices. As we have observed in Section 2.3.1, the (s,k)-polar partitions correspond to matrices of the kind before described, so the bound of the order also apply for the cograph minimal (s,k)-polar obstructions, being $(s+1)(k+1)$ for this graphs. Notice that for every positive integer s the bound below is attained since $2K_{s+1}$ is a cograph minimal $(s,1)$-polar obstruction.

We have from the previous lemma that every disconnected cograph minimal $(s,1)$-polar obstruction, except for $2P_3$ and $2K_{s+1}$, have only two connected components, and one of them is either an isolated vertex or it is isomorphic to K_2. The following two lemmas describe completely the structure of these cograph minimal $(s,1)$-polar obstructions; it is a bit surprising that for any integer s greater than or equal to 2, there are only two of such obstructions, one with a trivial component, and other with a two-vertex component.

Lemma 4.4. *Let s be an integer, $s \geq 2$, and let H be a connected cograph such that $G = H + K_2$ is a cograph minimal $(s,1)$-polar obstruction. Then G is isomorphic to $K_2 + (\overline{K_2} \oplus K_s)$.*



$K_2 + (\overline{K_2} \oplus K_s)$

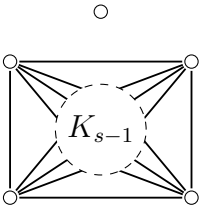
Figure 4.2: $K_2 + (\overline{K_2} \oplus K_s)$ is a cograph minimal $(s,1)$-polar obstruction.

Proof. Since G is a cograph $(s,1)$-polar obstruction, we have that H is not a complete s-partite graph, so H has K_{s+1} or $\overline{P_3}$ as induced subgraph. Nevertheless, if H has $\overline{P_3}$ as an induced subgraph, then $K_1 + 2K_2$ is an induced

subgraph of G, contradicting the minimality of G. Therefore H has K_{s+1} as induced subgraph.

Let K be a subset of $V(H)$ such that $H[K] \cong K_{s+1}$. Since H is $\overline{P_3}$-free, each vertex of H that is not in K is adjacent to every vertex in K, except maybe to one of them. Moreover, since G is an $(s,1)$-polar obstruction, H is not a complete graph, and in consequence there is a vertex v of H that is non-adjacent to at least one vertex in K. Note that $H[K \cup \{v\}]$ is isomorphic to $\overline{K_2} \oplus K_s$, and hence, G has $K_2 + (\overline{K_2} \oplus K_s)$ as an induced subgraph. But it is easy to verify that $K_2 + (\overline{K_2} \oplus K_s)$ is a cograph minimal $(s,1)$-polar obstruction, so, from the minimality of G we have that G is isomorphic to $K_2 + (\overline{K_2} \oplus K_s)$. □

Lemma 4.5. *Let s be an integer, $s \geq 2$, and let H be a connected cograph such that $G = H + K_1$ is a cograph minimal $(s,1)$-polar obstruction non isomorphic to $K_1 + (\overline{K_2} \oplus \overline{P_3})$. Then G is isomorphic to $K_1 + (C_4 \oplus K_{s-1})$.*



$K_1 + (C_4 \oplus K_{s-1})$

Figure 4.3: $K_1 + (C_4 \oplus K_{s-1})$ is a cograph minimal $(s,1)$-polar obstruction.

Proof. Since G is not a split graph, G has $2K_2$ or C_4 as an induced subgraph, and evidently these subgraphs must be induced subgraphs of H. If H has $2K_2$ as induced subgraph, then G contains $K_1 + 2K_2$ as an induced subgraph, and by the minimality of G, it must be isomorphic to $K_1 + 2K_2$, contradicting that G has only two connected components. So H has a cycle on four vertices, $C = (c_1, c_2, c_3, c_4)$, as an induced subgraph.

Let v a vertex of H that is not in C, which must exist, or else G would be $(2,1)$-polar, and hence $(s,1)$-polar. Then v must be adjacent to some vertex of C, otherwise G would have $\overline{K_2} + C_4$ as induced subgraph, which is not

possible. On the other hand, since G is a cograph, v cannot be adjacent to exactly one vertex of C nor can be adjacent to exactly two adjacent vertices of C. Furthermore, if v is adjacent to three vertices of C, then $V(C) \cup \{v\}$ induces $\overline{K_2} \oplus \overline{P_3}$, and therefore G has $K_1 + (\overline{K_2} \oplus \overline{P_3})$ as an induced subgraph, which by the minimality of G implies that G is isomorphic to $K_1 + (\overline{K_2} \oplus \overline{P_3})$, but we are assuming that G is not. So we have that every vertex of H that is not in C must be adjacent to every vertex of C or to exactly a pair of non adjacent vertices of C.

Let V_1, V_2 and V_3 be the subsets of vertices of H that are not vertices of C and that are adjacent to c_1 and c_3, to c_2 and c_4, and to c_i for every $i \in \{1,2,3,4\}$, respectively. Notice that every vertex in V_1 must be adjacent to every vertex of V_2; if x and y are non adjacent vertices in V_1 and V_2, respectively, then the path (y, c_4, c_1, x) is an induced path of G, contradicting that G is a cograph. Furthermore, any two vertices of V_1 are not adjacent, because if x and y are two adjacent vertices of V_1, then the graph induced by the set $\{c_1, c_3, c_4, x, y\}$ is isomorphic to $\overline{K_2} \oplus \overline{P_3}$, and then G contains $K_1 + (\overline{K_2} \oplus \overline{P_3})$ as induced subgraph. Analogously, any two vertices of V_2 are not adjacent.

Then, if $V_3 = \varnothing$, we have that H is a complete bipartite graph with bipartition $\{V_1 \cup \{c_2, c_4\}, V_2 \cup \{c_1, c_3\}\}$, and hence G is an $(s,1)$-polar graph, which is impossible. Thus, it must be the case that $V_3 \neq \varnothing$. Moreover, notice that every vertex in V_3 is adjacent to every vertex of V_1; if x, y are non adjacent vertices of V_1 and V_3, respectively, then the set of $\{c_1, c_3, c_4, x, y\}$ induces $\overline{K_2} \oplus \overline{P_3}$ in H, and then G contains $K_1 + (\overline{K_2} \oplus \overline{P_3})$ as an induced subgraph, which cannot be. Analogously, every vertex in V_3 is adjacent to every vertex in V_2.

Since every vertex in V_3 is adjacent to every vertex of $H - V_3$, and $H - V_3$ is a complete bipartite graph, if $H[V_3]$ was a complete $(s-2)$-partite graph then H would be a complete s-partite graph, and therefore G would be an $(s,1)$-polar graph. Thus, since we are assuming that G is a $(s,1)$-polar obstruction, V_3 cannot induce a complete $(s-2)$-partite graph, and in consequence $H[V_3]$ has $\overline{P_3}$ or K_{s-1} as an induced subgraph.

Nevertheless, we claim that the graph induced by V_3 is a $\overline{P_3}$-free graph. Otherwise, if $H[V_3]$ has $\overline{P_3}$ as an induced subgraph, then, together with any two non adjacent vertex of C this would induce a $\overline{K_2} \oplus \overline{P_3}$, which cannot occur. Then $H[V_3]$ has K_{s-1} as an induced subgraph, and hence G has $K_1 + (C_4 \oplus K_{s-1})$ as induced subgraph. But, as can be easily checked, $K_1 + (C_4 \oplus K_{s-1})$ is a cograph minimal $(s,1)$-polar obstruction, so G is isomorphic to $K_1 + (C_4 \oplus K_{s-1})$. $\square$

So far, we have characterized all disconnected cograph minimal $(s,1)$-polar obstructions, which are a constant number for any choice of s: four essential obstructions, $2K_{s+1}$, $K_2+(\overline{K_2}\oplus K_s)$ and $K_1+(C_4\oplus K_{s-1})$. Taking into account that the number of minimal $(s,0)$-polar obstructions is two, regardless of the choice of s, it would seem possible to have a constant number of cograph minimal $(s,1)$-polar obstructions, we would only need to show that the number of such connected obstructions is a constant independent from s. Unfortunately, we will show that this is not the case.

4.2.2 Connected obstructions

It is easy to verify that a cograph G is a connected minimal $(s,1)$-polar obstruction if and only if $\overline{G}$ is a disconnected minimal $(1,s)$-polar obstruction. Thus, in order to characterize the connected cograph minimal $(s,1)$-polar obstructions, we will study their complements, the disconnected cograph minimal $(1,k)$-polar obstructions. Notice that a graph G is a $(1,k)$-polar graph if and only if there exists a partition $\{A,B\}$ of $V(G)$ such that B induces a k-cluster and $G-A$ is an empty graph. In the following lemma we prove that the disconnected cograph minimal $(1,k)$-polar obstructions have a limited number of connected components.

Lemma 4.6. *Let k be a non negative integer, and let G be a cograph minimal $(1,k)$-polar obstruction, $G\ncong(k+1)K_2$. Then G has at most k components, and each of them is nontrivial.*

Proof. Suppose for a contradiction that G has an isolated vertex v. Since G is a cograph minimal $(1,k)$-polar obstruction, $G-v$ admits a $(1,k)$-polar partition $\{A,B\}$, but in such case $\{A\cup\{v\},B\}$ is a $(1,k)$-polar partition of G, contradicting the minimality of G. Thus, we conclude that every component of G has at least two vertices.

On the other hand, $H=(k+1)K_2$ is a cograph $(1,k)$-polar obstruction, because every k-cluster K of H intersects at most k components of H, and therefore $H-K$ is a nonempty graph. Furthermore for every vertex $v\in V(H)$, $H-v$ is isomorphic to kK_2+K_1, which is clearly a $(1,k)$-polar cograph, so H is a cograph minimal $(1,k)$-polar obstruction.

Finally, if G has more than k components, since none of them is an isolated vertex, G has $(k+1)K_2$ as an induced subgraph, so that $G\cong(k+1)K_2$. Thus, if $G\ncong(k+1)K_2$, then G has at most k components. □

At following, we will obtain a recursive characterization of disconnected cograph minimal $(1,k)$-polar obstructions. We begin by describing a construction of cograph minimal $(1,k)$-polar obstructions as disjoint union of smaller minimal polar obstructions.

Lemma 4.7. *Let t be an integer, $t \geq 2$, and for each $i \in \{1,\ldots,t\}$, let G_i be a connected cograph minimal $(1,k_i)$-polar obstruction that is a $(1,k_i+1)$-polar graph. Then, for $m = t-1+\sum_{i=1}^{t} k_i$, the graph $G = G_1 + \ldots + G_t$ is a cograph minimal $(1,m)$-polar obstruction that is a $(1,m+1)$-polar graph.*

Proof. Let $G_1,\ldots,G_t$ and G be as in the hypothesis. We first prove by means of a contradiction that G is a cograph $(1,m)$-polar obstruction. Suppose that G admits a $(1,m)$-polar partition $\{A,B\}$, and define for each $i \in \{1,2,\ldots,t\}$ the sets $A_i = V(G_i) \cap A$ and $B_i = V(G_i) \cap B$. Observe that every component of $G[B]$ is contained in a component of G. Denote the number of components of $G_i[B_i]$ by l_i; then, if $k_i < l_i$ for every $i \in \{1,\ldots,t\}$, we would have that $m+1 = \sum_{i=1}^{t}(k_i+1) \leq \sum_{i=1}^{t} l_i \leq m$, a contradiction. Nevertheless, we have that G_i is a cograph $(1,k_i)$-polar obstruction and $\{A_i,B_i\}$ is a $(1,l_i)$-polar partition of G_i, so that $k_i < l_i$ for every $i \in \{1,\ldots,t\}$, contradicting our previous argument. Since the contradiction arises from assuming that G is a $(1,m)$-polar cograph, we conclude that G is a cograph $(1,m)$-polar obstruction. Now we prove that G is minimal with this property.

If $v \in V(G)$, then $v \in V(G_j)$ for some $j \in \{1,\ldots,t\}$, say, without loss of generality, for $j=1$. Since G_1 is a cograph minimal $(1,k_1)$-polar obstruction, the graph $G_1 - v$ admits a $(1,k_1)$-polar partition $\{A_1,B_1\}$, and since by hypothesis G_i is a $(1,k_i+1)$-polar cograph for each $i \in \{2,3,\ldots,t\}$, we have that G_i admits a $(1,k_i+1)$-polar partition $\{A_i,B_i\}$. Therefore, $G-v$ is a $(1,m)$-polar cograph with polar partition $\{\bigcup_{i=1}^{t} A_i, \bigcup_{i=1}^{t} B_i\}$. Thus, G is a cograph minimal $(1,m)$-polar obstruction.

Finally, since for each $i \in \{1,2,\ldots,t\}$ the graph G_i admits a $(1,k_i+1)$-polar partition $\{A_i,B_i\}$, then $\{\bigcup_{i=1}^{t} A_i, \bigcup_{i=1}^{t} B_i\}$ is a $(1,m+1)$-polar partition of G, and therefore G is a $(1,m+1)$-polar cograph. □

Our goal is to prove that the cographs described in Lemma 4.7 are the only disconnected cograph minimal $(1,k)$-polar obstructions. In order to achieve this, we need the following technical, yet simple, result.

Lemma 4.8. *Let t be an integer, $t \geq 2$, and for each $i \in \{1,\ldots,t\}$, let G_i be a connected cograph minimal $(1,k_i)$-polar obstruction that is a $(1,k_i+1)$-polar*

graph. Then, for $m = t - 1 + \sum_{i=1}^{t} k_i$ *and for any non negative integer* μ, $\mu < m$, *the graph* $G = G_1 + \ldots + G_t$ *is not a cograph minimal* $(1,\mu)$*-polar obstruction.*

Proof. By considering the different cases in the characterization of disconnected cograph minimal $(s,1)$-polar obstructions, it is not hard to verify that any connected cograph minimal $(1,s)$-polar obstruction G that is $(1,s+1)$-polar contains, for any non negative integer σ such that $\sigma < s$, a proper induced subgraph G' that is both, a cograph minimal $(1,\sigma)$-polar obstruction and a $(1,\sigma+1)$-polar graph.

Let μ be a positive integer such that $\mu < m$, and let $s_1, \ldots, s_t$ be integers such that, for $i \in \{1,\ldots,t\}$, $0 \leq s_i \leq k_i$ and $\mu = t - 1 + \sum_{i=1}^{t} s_i$. By the choice of μ, $s_i < k_i$ for at least one $i \in \{1,\ldots,t\}$. For each $i \in \{1,\ldots,t\}$, if $s_i < k_i$ let H_i be a proper induced subgraph of G_i that is both, a cograph minimal $(1,s_i)$-polar obstruction and a $(1,s_i+1)$-polar graph, otherwise let $H_i = G_i$. Then, by Lemma 4.7, $H = H_1 + \cdots + H_t$ is a cograph minimal $(1,\mu)$-polar obstruction that is a proper induced subgraph of G, and therefore G is not a cograph minimal $(1,\mu)$-polar obstruction. □

We conclude the analysis of the disconnected cograph minimal $(1,k)$-polar obstructions by showing that the cographs described in Lemma 4.7 are the only ones.

Lemma 4.9. *Let* G *be a disconnected cograph minimal* $(1,k)$*-polar obstruction with components* $G_1, \ldots, G_t$. *Then, there exist non negative integers* $k_1, \ldots, k_t$ *such that for each* $i \in \{1,\ldots,t\}$, G_i *is a connected cograph minimal* $(1,k_i)$*-polar obstruction that is a* $(1,k_i+1)$*-polar cograph, and* $\sum_{i=1}^{t} k_i = k - t + 1$.

Proof. Since G is a cograph minimal $(1,k)$-polar obstruction we have that, for each $i \in \{1,\ldots,t\}$, the component G_i of G is a $(1,k)$-polar graph. For each $i \in \{1,\ldots,t\}$ and each $v \in V(G_i)$, let k_v be the minimum non negative integer such that $G_i - v$ is a $(1,k_v)$-polar graph, and let k_i be the maximum of k_v on all the vertices v of G_i, that is, $k_i = \max\{k_v \mid v \in V(G_i)\}$. Note that for each $i \in \{1,\ldots,t\}$ and any $v \in V(G_i)$, $G_i - v$ is a $(1,k_i)$-polar graph.

Moreover, we claim that for each i, $1 \leq i \leq t$, the graph G_i is not $(1,k_i)$-polar. Suppose for a contradiction that for some $i \in \{1,\ldots,t\}$, G_i is a $(1,k_i)$-polar graph, we will assume $i = 1$ without loss of generality. Let $\{X_1, Y_1\}$ be a $(1,k_1)$-polar partition of G_1, and let $v \in V(G_1)$ such that $G_1 - v$ is $(1,k_1)$-polar but it is not $(1,k_1-1)$-polar. Let $\{A,B\}$ be a $(1,k)$-polar partition of $G-v$. For

every $i \in \{1,\ldots,t\}$ define A_i and B_i in the following way, $A_1 = A \cap V(G_1 - v)$, $B_1 = B \cap V(G_1 - v)$, and for each $j \in \{2,\ldots,t\}$, let $A_j = A \cap V(G_j)$ and $B_j = B \cap V(G_j)$. Then $\{(A \setminus A_1) \cup X_1, (B \setminus B_1) \cup Y_1\}$ is a $(1,k)$-polar partition of G, a contradiction.

Thus, for each $i \in \{1,\ldots,t\}$, G_i is a connected cograph minimal $(1,k_i)$-polar obstruction that is $(1,k)$-polar, and in consequence $\overline{G_i}$ is a disconnected cograph minimal $(k_i,1)$-polar obstruction that is a $(k,1)$-polar graph. Observe that by Lemmas 4.2 to 4.5 this implies that $\overline{G_i}$ is one of $2K_{k_i+1}, K_2 + (\overline{K_2} \oplus K_{k_i})$ or $K_1 + (C_4 \oplus K_{k_i-1})$, and then, $\overline{G_i}$ is a disconnected cograph minimal $(k_i,1)$-polar obstruction that is $(k_i+1,1)$-polar. Equivalently, we have that G_i is a connected cograph minimal $(1,k_i)$-polar obstruction that is a $(1,k_i+1)$-polar graph.

Finally, by Lemmas 4.7 and 4.8 we have that, for $m = t - 1 + \sum_{i=1}^{t} k_i$, G is a cograph minimal $(1,m)$-polar obstruction that is a $(1,m+1)$-polar graph, and that G is not a cograph minimal $(1,\mu)$-polar obstruction for any integer μ with $0 \leq \mu < m$. Thus, since we are assuming that G is a cograph minimal $(1,k)$-polar obstruction, we have that $k = m$ and the result follows. □

4.2.3 Main results

Now, we can finally state our main result. Its proof is an immediate consequence of all previous lemmas.

Theorem 4.10. *Let G be a graph, and let s be an integer, $s \geq 2$. Then G is a cograph minimal $(s,1)$-polar obstruction if and only if it satisfies one of the following sentences:*

- *G is one of the four essential obstructions depicted in Figure 4.1, i.e., $K_1 + 2K_2, \overline{K_2} + C_4, 2P_3$ or $K_1 + (\overline{P_3} \oplus \overline{K_2})$.*
- *G is one of the four graphs $2K_{s+1}, K_2 + (\overline{K_2} \oplus K_s), K_1 + (C_4 \oplus K_{s-1})$ or $\overline{(s+1)K_2}$.*
- *The complement of G is disconnected with components $G_1,\ldots,G_t$, such that $t \leq s$, each G_i is the complement of a non-essential disconnected cograph minimal $(s_i,1)$-polar obstruction and $\sum_{i=1}^{t} s_i = s - t + 1$.*

To finish this section, we will prove that the four essential obstructions in Figure 4.1 constitute the family of minimal forbidden induced subgraphs for a cograph to admit an $(s,1)$-polar partition for some integer s, $s \geq 2$.

Lemma 4.11. *Let s be an integer. If G is a cograph minimal $(s,1)$-polar obstruction that is not essential, then the order of G is at least $s+1$.*

Proof. We will proceed by mathematical induction on s. The unique cograph minimal $(s,1)$-polar obstruction for $s=0$ is $2K_1$, while the unique two cograph minimal $(1,1)$-polar obstructions are C_4 and $2K_2$. This deals with the base case.

Let s be an integer, $s \geq 2$, and suppose that for every integer N such that $N < s$, if H is a non-essential cograph minimal $(N,1)$-polar obstruction, then H has at least $N+1$ vertices.

Let G be a non-essential cograph minimal $(s,1)$-polar obstruction. Observe that if G is disconnected, then by Lemmas 4.3, 4.4 and 4.5, the order of G is strictly greater than s. Else, G is a connected cograph and its complement, $\overline{G}$, is a disconnected cograph minimal $(1,s)$-polar obstruction; Lemmas 4.6 and 4.9 imply that either $\overline{G}$ is isomorphic to $(s+1)K_2$, which clearly has strictly more than $s+1$ vertices, or the components of $\overline{G}$ are $G_1,\ldots,G_t$ for some integer $t \in \{2,\ldots,s\}$, where each G_i is a cograph minimal $(1,s_i)$-polar obstruction for $1 \leq i \leq t$ and some non negative integer s_i, and $\sum_{i=1}^{t} s_i = s-t+1$. However, in the latter case we have by induction hypothesis that for every $i \in \{1,\ldots,t\}$, the order of G_i is at least s_i+1, which implies that

$$\begin{aligned} |\overline{G}| &= |G_1| + \cdots + |G_t| \\ &\geq (s_1+1) + \cdots + (s_t+1) \\ &= s_1 + \cdots + s_t + t \\ &= s - t + 1 + t \\ &= s+1, \end{aligned}$$

which ends the proof. □

Theorem 4.12. *Let G be a cograph. Then G admits an $(s,1)$-polar partition for some $s \geq 2$ if and only if it does not contain any of the essential obstructions (Figure 4.1) as an induced subgraph.*

Proof. Let G be a cograph such that for every integer s, $s \geq 2$, G is not an $(s,1)$-polar cograph. Particularly, G is not a $(n,1)$-polar cograph, where n stands for the order of G, and therefore G contains a cograph minimal $(n,1)$-polar obstruction H as induced subgraph. If H is not essential, then, by Lemma 4.11 we have that H has order at least $n+1$, which is impossible since H is a subgraph of G. Thus G contains an essential obstruction as an induced subgraph. The converse implication follows directly from Lemma 4.2. □

4.2.4 On the number of obstructions

Taking into consideration the number of cograph minimal $(s,1)$-polar obstructions for $s \in \{0,1,2\}$, it would seem that the number of these obstructions does not grow too fast. We finish this chapter by showing a quick estimation for the growth rate of the families of minimal obstructions; we prove that such a growth rate is sub exponential at best, and we exhibit an upper bound (with an extremely bad overestimation).

Let s be an integer, $s \geq 2$. In view of Lemmas 4.2 to 4.5, there are exactly seven disconnected cograph minimal $(s,1)$-polar obstructions, namely $2K_{s+1}$, $K_1 + (C_4 \oplus K_{s-1})$, $K_2 + (\overline{K_2} \oplus K_s)$, and the four essential obstructions depicted in Figure 4.1. Observe that the complements of the first three graphs mentioned above are the unique connected cograph minimal $(1,s)$-polar obstructions that are $(1,s+1)$-polar cographs.

On the other hand, to count the number of disconnected cograph minimal $(1,s)$-polar obstructions is equivalent to count the number of connected cograph minimal $(s,1)$-polar obstructions. Furthermore, by Lemma 4.9, each disconnected cograph minimal $(1,s)$-polar obstruction G with components $G_0,\ldots,G_k$ satisfies that G_i is a connected cograph minimal $(1,s_i)$-polar obstruction that is a $(1,s_i+1)$-polar cograph for each $i \in \{0,\ldots,k\}$, with $s = s_0 + \cdots + s_k + k$ where each term is a non negative integer. Since there is exactly one connected cograph minimal $(1,s_i)$-polar obstruction for $s_i \in \{0,1\}$, and there are exactly three of them which are connected for $s_i \geq 2$ we have the following.

Proposition 4.13. *Let s be an integer, $s \geq 2$. If s is expressed as a sum of non negative integers, $s = s_0 + s_1 + \cdots + s_k + k$, and there are exactly n of the terms s_i greater than 1, then there are at most 3^n non isomorphic disconnected cograph minimal $(1,s)$-polar obstructions G with connected components $G_0,\ldots,G_k$ such that G_i is a cograph minimal $(1,s_i)$-polar obstruction for each $i \in \{0,\ldots,k\}$.*

Let s be a non negative integer, and let $D(s)$ be the number of distinct ways in which s can be expressed as a sum $s = s_0 + s_1 + \cdots + s_k + k$, where $k \geq 1$ and s_i is a non negative integer for each $i \in \{0,1\ldots,k\}$, and where we are considering two of these representations of s as the same when they correspond to a permutation of the terms s_i. Thus, the preceding lemma gives straightforward bounds for the number of disconnected cograph minimal $(1,s)$-polar cographs in terms of $D(s)$.

Theorem 4.14. *Let s be an integer, $s \geq 2$. Then the number of disconnected cograph minimal $(1,s)$-polar obstructions, $n(s)$, is such that*

$$D(s) \leq n(s) \leq 3^m \cdot D(s) < 3^{s/2} \cdot D(s),$$

where m is the maximum possible number of terms s_i greater than one in a decomposition $s = s_0 + \cdots + s_k + k$ of s with $k \geq 1$.

Proof. The left inequality is due to the fact that for each decomposition of s as a sum of non negative integers $s = s_0 + \cdots + s_k + k$, there is at least one disconnected cograph minimal $(1,s)$-polar obstruction. The inequality in the middle is a direct consequence of Lemma 4.13, while the last inequality follows from the trivial fact that $m < s/2$. □

Let s be a non negative integer and let k be a positive integer. Observe that, if s can be expressed as a sum of non negative integers $s = s_0 + s_1 + \cdots + s_k + k$, then $s+1$ is the sum of the positive integers $s'_1, \ldots, s'_k$, where $s'_i = s_i + 1$ for each $i \in \{0, 1, \ldots, k\}$. Conversely, if a positive integer s' is written as a sum of positive integers $s'_0, \ldots, s'_k$, then $s' - 1$ can be expressed as the sum $s_0 + s_1 + \cdots + s_k + k$, where $s_i = s'_i - 1$ for each $i \in \{0, 1, \ldots, k\}$. Therefore, the number of distinct ways in which a non negative integer s can be expressed as a sum $s = s_0 + \cdots + s_k + k$, where k is a non-fixed positive integer and $s_0, s_1, \ldots, s_k$ are non negative integers, is equal to the number of distinct ways in which $s+1$ can be written as a sum of, at least two, positive integers.

Let n be a positive integer. The number of distinct ways in which n can be written as a sum of positive integers (including the expression of n as the sum of the unique term n) is called the n-th partition number. Observe that the observation in the previous paragraph directly relates $D(s)$ with the partition number of $s+1$ by means of the equation $D(s) = p(s+1) - 1$.

The partition number has been extensively studied, and particularly, Hardy and Ramanujan gave in 1918 the following asymptotic approximation.

Theorem 4.15. *[72] Let $p(n)$ be the number of ways of writing the positive integer n as a sum of positive integers, where the order of the terms is not considered. Then*

$$p(n) \sim \frac{1}{4n\sqrt{3}} \exp\left(\pi\sqrt{\frac{2n}{3}}\right).$$

Conclusions

The problem of characterizing subclasses of cographs given by a hereditary property by means of forbidden induced subgraphs raised in 1990 with the results of Damaschke [38]. Independently, the results given by Feder, Hell and, Xie in [57] imply that, for fixed non negative integers s and k, any hereditary subclass of (s,k)-polar graphs can be characterized by a finite family of induced subgraphs.

Characterization of (s,k)-polar cographs by forbidden induced subgraphs falls into both of these topics. Exact list of cograph minimal (s,k)-polar obstructions are known only for very small values of s and k. Ekim, Mahadev and de Werra were pioneers in the study of polar cographs; they provided a characterization by forbidden subgraphs for this class of graphs, as well as for the (s,k)-polar cographs when $\min\{s,k\} \leq 1$ [47]. Bravo, Nogueira, Protti and Vianna exhibited the complete family of forbidden induced subgraphs for the $(2,1)$-polar cographs, where it can be obtained, by simple complementation, analogous results for $(1,2)$-polar cographs . Recently, the problem of finding a characterization by forbidden induced subgraphs by the family of $(2,2)$-polar cographs, originally proposed in [47], was solved for Hell, Hernández-Cruz and Linhares-Sales. Characterizations before are all the known results on complete lists of cograph minimal (s,k)-polar obstructions, that is to say, these lists are known only when $0 \leq s,k \leq 2$, and when $\min\{s,k\} \leq 1$.

As a natural continuation of the previous lines of investigation, in this work we focused on the problem of characterizing $(s,1)$-polar cographs by families of forbidden induced subgraphs. Our main results, presented in Chapter 4, include the complete list of four cograph minimal $(s,1)$-polar obstructions for the case $s \geq 2$ (Lemma 4.2 and Theorem 4.12), as well as a bit surprising recursive characterization of the minimal forbidden induced subgraphs for the $(s,1)$-polar cographs (Theorem 4.10). In addition, we presented lower and upper bounds for the growth rate of the families of forbidden of minimal obstructions

(Theorem 4.14).

Known results suggest that the families of minimal obstructions for the (s,k)-polar cographs could be too large for giving exhaustive lists of its elements for arbitrary values of s and k. Nonetheless, our recursive characterization makes us wonder whether a similar result may be achieved for any values of s and k. Some encouraging partial results for the case $s = k$ show that maybe a combination of recursion together with a classification of some families of minimal obstructions may cover the whole family of minimal obstructions. Also, taking into account the results in [47] and Theorem 4.12, it seems possible to find the complete list of minimal obstructions to the problem of recognizing (s,t)-polar cographs, for some integer t and a fixed integer s, $s \geq 2$.

Another two possible future lines of work are the following. First, in view of the mentioned result of Damaschke [38], we have the problem of finding the families of minimal obstructions for any hereditary property on cographs. In this text was worked this problem for the particular case of graphs that have an $(s,1)$-polar partition, but several other hereditary properties may result interesting for its study. Particularly, we propose the study of cographs that admit some kind of M-partition as can be, for example, graphs that admits a partition into a complete s-partite graph and a complete k-partite graph, which correspond to an M-partition where the A and B blocks of the matrix M are matrices of size s and k, respectively, with 0's on its main diagonals and 1's anywhere else (or more generally, graphs that can be partitioned into a limited amount of complete multipartite graphs with restricted numbers of parts).

Second, in the context of the results obtained by Feder, Hell and Xie [57], we propose to work the characterization by means of forbidden induced subgraphs of some subclasses of (s,k)-polar graphs. Particularly, we propose the work with the intersection of (s,k)-polar graphs and some superclasses of cographs, as can be the P_4-sparse graphs (graphs such that every set of five vertices contains at most one induced P_4), or the P_4-reducible graphs (graphs for which any vertex is contained in at most one P_4). Both, P_4-sparse and P_4-reducible graphs, are families of graphs with a property of *low local density*, which arises in many application of the real life (see [34] for some examples).

Bibliography

[1] Alon N., Krivelevich, M., & Sudakov, B. (1997). Subgraphs with a large cochromatic number. *Journal of Graph Theory, 25*(4), 295-297.

[2] Arditti, J. C., & de Werra, D. (1976). A note on a paper by D. Seinsche. *Journal of Combinatorial Theory, Series B, 21*(1), 90.

[3] Arnborg, S., Lagergren, J., & Seese, D. (1991). Easy problems for tree-decomposable graphs. *Journal of Algorithms, 12*(2), 308-340.

[4] Bandelt, H. J., & Mulder, H. M. (1986). Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B, 41*(2), 182-208.

[5] Berge, C. (1960). Les problemes de coloration en théorie des graphes. *Publ. Inst. Stat. Univ. Paris, 9*, 123-160.

[6] Berge, C., & Chvátal, V. (Eds.). (1984). *Topics on perfect graphs* (Vol. 21). Elsevier.

[7] Bodlaender, H. L., & Möhring, R. H. (1993). The pathwidth and treewidth of cographs. *SIAM Journal on Discrete Mathematics, 6*(2), 181-188.

[8] Bondy, J. A., & Murty, U. S. (2008). *Graph theory*. Graduate texts in mathematics, 1(244), ALL-ALL.

[9] Bose, P., Buss, J. F., & Lubiw, A. (1998). Pattern matching for permutations. *Information Processing Letters, 65*(5), 277-283.

[10] Brandstädt, A. (1996). Partitions of graphs into one or two stable sets and cliques. *Discrete Mathematics, 152*, 47-54.

[11] Brandstädt, A. (1998). Corrigendum. *Discrete Mathematics, 186*, 295.

[12] Brandstädt, A., Le, V. B., & Spinrad, J. P. (1999). *Graph Classes: a Survey*, volume 3 of SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia.

[13] Brandstädt, A., & Szymczak, T. (1998). The complexity of some problems related to graph 3-colorability. *Discrete Applied Mathematics, 89*(1-3), 59-73.

[14] Bravo, R., Nogueira, L., Protti, F. & Vianna, C., Obstruções Minimais de Cografos-$(2, 1)$ com restrição externa, in: Annals of I ETC - Encontro de Teoria da Computação (CSBC 2016) (2016) Porto Alegre, `http://ebooks.pucrs.br/edipucrs/anais/csbc`, ISSN 2175-2761 (In Portuguese).

[15] Bretscher, A., Corneil, D., Habib, M., & Paul, C. (2008). A simple linear time LexBFS cograph recognition algorithm. *SIAM Journal on Discrete Mathematics, 22*(4), 1277-1296.

[16] Broere, I., & Burger, M. (1989). Critically cochromatic graphs. *Journal of graph theory, 13*(1), 23-28.

[17] Broersma, H. J., Dahlhaus, E., & Kloks, T. (2000). A linear time algorithm for minimum fill-in and treewidth for distance hereditary graphs. *Discrete Applied Mathematics, 99*(1-3), 367-400.

[18] Brooks, R. (1941). On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society, 37*(2), 194-197.

[19] Burlet, M., & Uhry, J. P. (1984). *Parity graphs*. In North-Holland mathematics studies (Vol. 88, pp. 253-277). North-Holland.

[20] Capelle, C., Cournier, A., & Habib, M. (1994). Cograph recognition algorithm revisited and online induced P4 search. *LIRMM, Universite Montpellier, rapport de recherche, 94073*.

[21] Chartrand, G., & Oellermann, O. R. (1993). *Applied and algorithmic graph theory* (Vol. 993). New York: McGraw-Hill.

[22] Chartrand, G., & Zhang, P. (2008). *Chromatic graph theory*. CRC press.

[23] Chernyak, Z. A., & Chernyak, A. A. (1986). About recognizing (α, β) classes of polar graphs. *Discrete Mathematics, 62*(2), 133-138.

[24] Christen, C. A., & Selkow, S. M. (1979). Some perfect coloring properties of graphs. *Journal of Combinatorial Theory, Series B, 27*(1), 49-59.

[25] Chudnovsky, M., Cornuéjols, G., Liu, X., Seymour, P., & Vušković, K. (2005). Recognizing Berge Graphs. *Combinatorica, 2*(25), 143-186.

[26] Chudnovsky, M., Robertson, N., Seymour, P., & Thomas, R. (2006). The strong perfect graph theorem. *Annals of mathematics*, 51-229.

[27] Churchley, R., & Huang, J. (2011). Line-polar graphs: characterization and recognition. *SIAM Journal on Discrete Mathematics, 25*(3), 1269-1284.

[28] Churchley, R., & Huang, J. (2012). List monopolar partitions of claw-free graphs. *Discrete Mathematics, 312*(17), 2545-2549.

[29] Churchley, R., & Huang, J. (2014). On the polarity and monopolarity of graphs. *Journal of Graph Theory, 76*(2), 138-148.

[30] Churchley, R., & Huang, J. (2014). Solving partition problems with colour-bipartitions. *Graphs and Combinatorics, 30*(2), 353-364.

[31] Chvátal, V. (1984). *Perfectly ordered graphs*. In North-Holland mathematics studies (Vol. 88, pp. 63-65). North-Holland.

[32] Cook, S. A. (1971, May). The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing* (pp. 151-158). ACM.

[33] Corneil, D. G., Lerchs, H., & Burlingham, L. S. (1981). Complement reducible graphs. *Discrete Applied Mathematics, 3*(3), 163-174.

[34] Corneil, D. G., Perl, Y., & Stewart, L. K. (1985). A linear recognition algorithm for cographs. *SIAM Journal on Computing, 14*(4), 926-934.

[35] Courcelle, B. (1992). The monadic second-order logic of graphs III: Tree-decompositions, minors and complexity issues. *RAIRO-Theoretical Informatics and Applications, 26*(3), 257-286.

[36] Courcelle, B., Makowsky, J. A., & Rotics, U. (2000). Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems, 33*(2), 125-150.

[37] Courcelle, B., & Olariu, S. (2000). Upper bounds to the clique width of graphs. *Discrete Applied Mathematics, 101*(1-3), 77-114.

[38] Damaschke, P. (1990). Induced subgraphs and well-quasi-ordering. *Journal of Graph Theory, 14*(4), 427-435.

[39] Damaschke, P. (1991, June). *Induced subraph isomorphism for cographs is NP-complete*. In International Workshop on Graph-Theoretic Concepts in Computer Science (pp. 72-78). Springer, Berlin, Heidelberg.

[40] Demange, M., Ekim, T., & De Werra, D. (2005). Partitioning cographs into cliques and stable sets. *Discrete Optimization, 2*(2), 145-153.

[41] Diestel, R. (2005). *Graph theory* (Graduate texts in mathematics) (Vol. 173). Heidelberg: Springer.

[42] Ekim, T., & de Werra, D. (2005). On split-coloring problems. *Journal of Combinatorial Optimization, 10*(3), 211-225.

[43] Ekim, T., Heggernes, P., & Meister, D. (2009, June). Polar permutation graphs. In *International Workshop on Combinatorial Algorithms* (pp. 218-229). Springer, Berlin, Heidelberg.

[44] Ekim, T., Heggernes, P., & Meister, D. (2013). Polar permutation graphs are polynomial-time recognisable. *European Journal of Combinatorics, 34*(3), 576-592.

[45] Ekim, T., Hell, P., Stacho, J., & de Werra, D. (2008). Polarity of chordal graphs. *Discrete Applied Mathematics, 156*(13), 2469-2479.

[46] Ekim, T., & Huang, J. (2010). Recognizing line-polar bipartite graphs in time $O(n)$. *Discrete Applied Mathematics, 158*(15), 1593-1598.

[47] Ekim, T., Mahadev, N. V., & de Werra, D. (2008). Polar cographs. *Discrete Applied Mathematics, 156*(10), 1652-1660.

[48] Ekim, T., Mahadev, N. V., & de Werra, D. (2014). Corrigendum to "Polar cograph" [Discrete Appl. Math. 156 (2008) 1652–1660]. *Discrete Applied Mathematics, 171*, 158.

[49] Erdös, P., & Gimbel, J. (1993). Some problems and results in cochromatic theory. In *Annals of Discrete Mathematics* (Vol. 55, pp. 261-264). Elsevier.

[50] Erdös, P., Gimbel, J., & Kratsch, D. (1991). Some extremal results in cochromatic and dichromatic theory. *Journal of Graph Theory, 15*(6), 579-585.

[51] Erdös, P., Gimbel, J., & Straight, H. J. (1990). Chromatic number versus cochromatic number in graphs with bounded clique number. *European Journal of Combinatorics, 11*(3), 235-240.

[52] Eschen, E. M., & Wang, X. (2014). Algorithms for unipolar and generalized split graphs. *Discrete Applied Mathematics, 162*, 195-201.

[53] Farber, M., & Keil, J. M. (1985). Domination in permutation graphs. *Journal of algorithms, 6*(3), 309-321.

[54] Farrugia, A. (2004). Vertex-partitioning into fixed additive induced-hereditary properties is NP-hard. *The electronic journal of combinatorics, 11*(1), 46.

[55] Feder, T., Hell, P., Klein, S., & Motwani, R. (1999). Complexity of graph partition problems. In *Proc. 31st Annu. ACM Symp. on Theory of Computing* (pp. 464-472).

[56] Feder, T., Hell, P., Klein, S., & Motwani, R. (2003). List partitions. *SIAM Journal on Discrete Mathematics, 16*(3), 449-478.

[57] Feder, T., Hell, P., & Xie, W. (2007). Matrix partitions with finitely many obstructions. *Electronic Notes in Discrete Mathematics, 28*, 371-378.

[58] Foldes, S., & Hammer, P. L. (1976). *Split graphs*. Universität Bonn. Institut für Ökonometrie und Operations Research.

[59] Foulis, D. J. (1969). Empirical logic. In *Xeroxed course notes*. Univ. of Massachusetts.

[60] Fulkerson, D., & Gross, O. (1965). Incidence matrices and interval graphs. *Pacific Journal of Mathematics, 15*(3), 835-855.

[61] Gagarin, A. V. (1999). Chordal $(1, \beta)$-polar graphs, *Vestsi Nats. Akad. Navuk Belarusi Ser. Fiz. Mat. Navuk, 143*, 115-118.

[62] Garey, M. R., & Johnson, D. S. (2002). *Computers and intractability* (Vol. 29). New York: wh freeman.

[63] Ghoula-Houri A. (1962). Caracteérisation des graphes non orientes dont on peut orienter les aretes de maniere a obtenir le graphe d'une relation d'ordre, *C.R. Acad. Sci. Paris 254*, 1370-1371.

[64] Gimbel, J. (1984). *The chromatic and cochromatic number of a graph.* PhD. dissertation. Western Michigan University.

[65] Gimbel, J., Kratsch, D., & Stewart, L. (1994). On cocolourings and cochromatic numbers of graphs. *Discrete Applied Mathematics, 48*(2), 111-127.

[66] Gimbel, J., & Straight, H. J. (1987). Some topics in cochromatic theory. *Graphs and Combinatorics, 3*(1), 255-265.

[67] Golumbic, M. C. (1977). The complexity of comparability graph recognition and coloring. *Computing, 18*(3), 199-208.

[68] Golumbic, M. C. (2004). *Algorithmic graph theory and perfect graphs* (Vol. 57). Elsevier.

[69] Golumbic, M. C., & Rotics, U. (2000). On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science, 11*(03), 423-443.

[70] Grötschel, M., Lovász, L., & Schrijver, A. (2012). *Geometric algorithms and combinatorial optimization* (Vol. 2). Springer Science & Business Media.

[71] Hammer, P. L., & Simeone, B. (1981). The splittance of a graph. *Combinatorica, 1*(3), 275-284.

[72] Hardy, G. H., & Ramanujan, S. (1918). Asymptotic formulaæin combinatory analysis. *Proceedings of the London Mathematical Society, 2*(1), 75-115.

[73] Hell, P. (2014). Graph partitions with prescribed patterns. *European Journal of Combinatorics, 35*, 335-353.

[74] Hell, P., Hernández-Cruz, C., & Linhares-Sales, C. Minimal Obstructions to 2-polar partitions, *Discrete Applied Mathematics*, *accepted*.

[75] Hell, P., Klein, S., Nogueira, L. T., & Protti, F. (2004). Partitioning chordal graphs into independent sets and cliques. *Discrete Applied Mathematics, 141*(1-3), 185-194.

[76] Hopcroft, J. E., & Tarjan, R. E. (1972). Isomorphism of planar graphs. In *Complexity of computer computations* (pp. 131-152). Springer, Boston, MA.

[77] Howorka, E. (1977). A characterization of distance-hereditary graphs. *The quarterly journal of mathematics, 28*(4), 417-420.

[78] Huang, J., & Xu, B. (2010). A forbidden subgraph characterization of line-polar bipartite graphs. *Discrete Applied Mathematics, 158*(6), 666-680.

[79] Joeris, B., Lindzey, N., McConnell, R. M., & Osheim, N. (2017). Simple DFS on the complement of a graph and on partially complemented digraphs. *Information Processing Letters, 117*, 35-39.

[80] Jung, H. A. (1978). On a class of posets and the corresponding comparability graphs. *Journal of Combinatorial Theory, Series B, 24*(2), 125-133.

[81] Jungnickel, D., (1999).*Graphs, networks and algorithms* (Vol. 5). Berlin: Springer.

[82] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85-103). Springer, Boston, MA.

[83] Le, V. B. & Nevries, R. (2014). Complexity and algorithms for recognizing polar and monopolar graphs. *Theoretical Computer Science, 528*, 1-11.

[84] Le, V. B. & Nevries, R. (2011, December). Recognizing polar planar graphs using new results for monopolarity. In *International Symposium on Algorithms and Computation* (pp. 120-129). Springer, Berlin, Heidelberg.

[85] Lesniak, L., & Straight, H. J. (1977). The cochromatic number of a graph. *Ars Combinatoria, 3*, 39-46.

[86] Lovász, L. (1972). A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B, 13*(2), 95-98.

[87] Lovász, L. (1972). Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2(3), 253-267.

[88] Lovász, L. (1975). Three short proofs in graph theory. *Journal of Combinatorial Theory, Series B, 19*(3), 269-271.

[89] Meister, D. (2010). Treewidth and minimum fill-in on permutation graphs in linear time. *Theoretical Computer Science, 411*(40-42), 3685-3700.

[90] Mel'nikov, O., & Kozhich, P. P. (1985). Algorithms for recognizing the polarity of a graph with bounded parameters. *Izvestia Akad. Nauk BSSR, ser. Fiz. Mat. Nauk, 6*, 50-54.

[91] McDiarmid, C., & Yolov, N. (2015). Recognition of unipolar and generalised split graphs. *Algorithms, 8*(1), 46-59.

[92] Middendorf, M., & Pfeiffer, F. (1990). On the complexity of recognizing perfectly orderable graphs. *Discrete Mathematics, 80*(3), 327-333.

[93] Möhring, R. H. (1989). Computationally tractable classes of ordered sets. In *Algorithms and order* (pp. 105-193). Springer, Dordrecht.

[94] Nicolai, F., & Szymczak, T. (2001). Homogeneous sets and domination: A linear time algorithm for distance hereditary graphs. *Networks, 37*(3), 117-128.

[95] Pandu Rangan, V. K. C. (1993). Efficient transitive reduction of permutation graphs and its applications. *Computer Science and Informatics, 23*, 52-52.

[96] Parra, A., & Scheffler, P. (1997). Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics, 79*(1-3), 171-188.

[97] Rose, D. J., Tarjan, R. E., & Lueker, G. S. (1976). Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing, 5*(2), 266-283.

[98] Seinsche, D. (1974). On a property of the class of n-colorable graphs. *Journal of Combinatorial Theory, Series B, 16*(2), 191-193.

[99] Stacho, J. (2008). *Complexity of generalized colourings of chordal graphs.* PhD. dissertation. School of Computing Science-Simon Fraser University.

[100] Straight, H. J. (1979). Cochromatic number and the genus of a graph. *Journal of Graph Theory, 3*(1), 43-51.

[101] Straight, H. J. (1980). Note on the cochromatic number of several surfaces. *Journal of Graph Theory, 4*(1), 115-117.

[102] Sumner, D. P. (1974). Dacey graphs. *Journal of the Australian Mathematical Society, 18*(4), 492-502.

[103] Talmaciu, M., & Nechita, E. (2010). On Polar, Trivially Perfect Graphs. *International Journal of Computers Communications & Control, 5*(5), 939-945.

[104] Tyshkevich, R. I., & Chernyak, A. A. (1985). Decomposition of graphs. *Cybernetics, 21*(2), 231-242.

[105] Tyshkevich, R. I., & Chernyak, A. A. (1985). Algorithms for the canonical decomposition of a graph and recognizing polarity. *Izvestia Akad. Nauk BSSR, ser. Fiz.-Mat. Nauk, 6*, 16-23.

[106] Yan, J. H., Chen, J. J., & Chang, G. J. (1996). Quasi-threshold graphs. *Discrete Applied Mathematics, 69*(3), 247-255.

[107] Yolov, N. (2016). Polarity and Monopolarity of 3-colourable comparability graphs. *arXiv preprint arXiv:1604.00905*.

Glossary of symbols

$\cap$	Intersection
$\cup$	Union
$\lvert X\rvert$	Cardinality of the set X
$\setminus$	Set-theoretic difference
$\subseteq$	Subset

Index