On interval number in cycle convexity

Julio Araujo[1] Guillaume Ducoffe[2,3*] Nicolas Nisse[4†] Karol Suchan[5,6]

[1] *ParGO, Departamento de Matemática, Universidade Federal do Ceará, Brazil*
[2] *National Institute for Research and Development in Informatics, Romania*
[3] *The Research Institute of the University of Bucharest ICUB, Romania*
[4] *Université Côte d'Azur, Inria, CNRS, I3S, France*
[5] *Universidad Adolfo Ibáñez, Santiago, Chile*
[6] *AGH - University of Science and Technology, Krakow, Poland*

received 25th July 2017, revised 28th Mar. 2018, accepted 16th Apr. 2018.

Recently, Araujo et al. [Manuscript in preparation, 2017] introduced the notion of Cycle Convexity of graphs. In their seminal work, they studied the graph convexity parameter called hull number for this new graph convexity they proposed, and they presented some of its applications in Knot theory. Roughly, the *tunnel number* of a knot embedded in a plane is upper bounded by the hull number of a corresponding planar 4-regular graph in cycle convexity. In this paper, we go further in the study of this new graph convexity and we study the interval number of a graph in cycle convexity. This parameter is, alongside the hull number, one of the most studied parameters in the literature about graph convexities. Precisely, given a graph G, its *interval number* in cycle convexity, denoted by $in_{cc}(G)$, is the minimum cardinality of a set $S \subseteq V(G)$ such that every vertex $w \in V(G) \setminus S$ has two distinct neighbors $u, v \in S$ such that u and v lie in same connected component of $G[S]$, i.e. the subgraph of G induced by the vertices in S.

In this work, first we provide bounds on $in_{cc}(G)$ and its relations to other graph convexity parameters, and explore its behaviour on grids. Then, we present some hardness results by showing that deciding whether $in_{cc}(G) \leq k$ is NP-complete, even if G is a split graph or a bounded-degree planar graph, and that the problem is W[2]-hard in bipartite graphs when k is the parameter. As a consequence, we obtain that $in_{cc}(G)$ cannot be approximated up to a constant factor in the classes of split graphs and bipartite graphs (unless $P = NP$).

On the positive side, we present polynomial-time algorithms to compute $in_{cc}(G)$ for outerplanar graphs, cobipartite graphs and interval graphs. We also present fixed-parameter tractable (FPT) algorithms to compute it for $(q, q-4)$-graphs when q is the parameter and for general graphs G when parameterized either by the treewidth or the neighborhood diversity of G.

Some of our hardness results and positive results are not known to hold for related graph convexities and domination problems. We hope that the design of our new reductions and polynomial-time algorithms can be helpful in order to advance in the study of related graph problems.

Keywords: graph convexity; interval number; domination problems in graphs; complexity and algorithms.

*This work was supported by the Institutional research programme PN 1819 "Advanced IT resources to support digital transformation processes in the economy and society - RESINFO-TD" (2018), project PN 1819-01-01"Modeling, simulation, optimization of complex systems and decision support in new areas of IT&C research", funded by the Ministry of Research and Innovation, Romania.

†This work has been partially supported by ANR project Stint (ANR-13-BS02-0007), the ANR program "Investments for the Future" (ANR-11- LABX-0031-01), the associated Inria team AlDyNet.

ISSN 1365–8050

1 Introduction

A convexity in graphs can be roughly described as the rules of propagation of an infection process on the vertices. Several graph convexities have been defined in the literature with different practical and theoretical motivations Varlet (1976); Jamison (1982); Farber and Jamison (1986); Duchet (1988); Changat and Mathew (1999). We refer to Section 1.1 for more details. Apart from the study of a new graph convexity, the main contributions of this paper are the design of new hardness reductions and polynomial-time algorithms in order to tackle with convexity problems on graphs. We expect our results to be used for the study of other graph convexities than just the one described in this work (formal relationships between all these convexities are presented in Section 2.2).

There are several real-world applications that can be modelled by an infection process in a network. Many of these applications can be seen as graph convexity problems such as, *e.g.*, disseminated diseases Balogh and Pete (1998), spread of opinion and community formation in social networks Dreyer and Roberts (2009); Wasserman and Faust (1994) and distributed computing Peleg (2002). Depending on the application, some works in the literature name such processes as bootstrap percolation Balister et al. (2010); Balogh and Bollobás (2003), local majority processes Peleg (2002), catastrophic fault patterns Nayak et al. (2000), etc. Recently, a new graph convexity called *cycle convexity* has been defined for its applications in Knot Theory Araujo et al. (2018). Roughly, in the corresponding process we infect a vertex only if it belongs to a cycle where all other vertices are infected. Relationships between the *tunnel number* of a knot in the plane and the hull number of some planar graph in this new graph convexity have been proved in Araujo et al. (2018). Such topological aspects are out of the scope of the present paper. Nevertheless, we think that the study of cycle convexity can be interesting on its own, and for its relationships with more studied convexity and domination problems on graphs. This paper is dedicated to the study of the graph convexity parameter called *interval number* of graphs in this new graph convexity. Before going into the details of cycle convexity and of the corresponding interval number of a graph, let us recall general definitions of convexity and graph convexity.

A *convexity space* is an ordered pair $(V, \mathcal{C})$, where V is a non-empty set and $\mathcal{C}$ is a family of subsets of V, called *convex sets*, that satisfies:

(C1) $\emptyset, V \in \mathcal{C}$; and

(C2) $C \cap C' \in \mathcal{C}$, for all $C, C' \in \mathcal{C}$.

Given a subset $C \subseteq V$, the *convex hull of* C (with respect to $(V, \mathcal{C})$) is the unique inclusion wise minimal $C' \in \mathcal{C}$ containing C, and it is denoted by $hull_{(V,\mathcal{C})}(C)$. If $hull_{(V,\mathcal{C})}(C) = V$, then C is said to be a *hull set* of $(V, \mathcal{C})$. The *hull number of* V *with respect to* $\mathcal{C}$ is the size of a minimum hull set, and is denoted by $hn(V, \mathcal{C})$.

When studying Euclidean spaces, the set V is $\mathbb{R}^n$ and $\mathcal{C}$ contains all the convex subsets of $\mathbb{R}^n$. Recall that a set $C \subseteq \mathbb{R}^n$ is *convex* if, for any two points p_1 and p_2 in C, any convex combination of p_1 and p_2 also belongs to C. In other words, C is convex if all the points on the line segment between p_1 and p_2 also belong to C, for every pair of points $p_1, p_2 \in C$. Remember that the convex hull of a set of points C in $\mathbb{R}^n$ can be computed by recursively augmenting C with any point that lies on a segment between two points already in C. Metaphorically, it can be seen as a contagion (or percolation) process starting from the initial set C that infects (or percolates) the whole convex hull $hull_{(V,\mathcal{C})}(C)$.

These notions where brought to graph theory in the 80's Farber and Jamison (1986); Duchet (1988) by taking the vertex set of an input graph $G = (V, E)$ as the set V. The set $\mathcal{C}$ of convex sets depends on the definition of the "contagion process" that is defined in terms of an *interval function*. For more details, we refer the reader to Pelayo (2013). Let us now present some formal definitions concerning graph convexities and their parameters.

1.1 Related work

General approach. Let us first give a general definition of f-convexity according to some function f.

Let $G = (V, E)$ be a finite graph. An *interval function* $f : 2^V \times V \to 2^V$ takes a subset $X \subseteq V$, the set of *infected* vertices, and a vertex $v \in V \setminus X$, a *healthy* vertex, as an input and returns either $X \cup \{v\}$, in which case the vertex has been infected or X, when x remains healthy. That is, for any $X \subset V$ and $v \in V \setminus X$, $f(X, v) \in \{X, X \cup \{v\}\}$. If $v \in f(X, v)$, v is said to be *generated* by X. Furthermore we suppose f to be monotonic for every fixed vertex v, *i.e.* if v is generated by X then it is also generated by any superset $Y \supseteq X$.

For any $X \subseteq V$, let us define the *interval* $f(X)$ of X as $\bigcup_{v \in V \setminus X} f(X, v)$, i.e. $f(X)$ consists of X plus any vertex that is generated by X according to f. By the monotonicity of f, for every *fixed points* X, Y of f we have $f(X \cap Y) \subseteq f(X) \cap f(Y) \subseteq X \cap Y \subseteq f(X \cap Y)$. In particular, $X \cap Y$ is also a fixed point of f, and it holds $f(X \cap Y) = f(X) \cap f(Y)$. We can then define a convexity space $(V, \mathcal{C}_f)$ with $\mathcal{C}_f \subseteq 2^V$ containing the fixed points of f. The interval $f(X)$ of X is also called the set *generated* by X, and X is a *generator* of $f(X)$. We say that a set $X \subseteq V$ is a *generator* (a.k.a. *interval set*) for G if $f(X) = V$. A generator X of $f(X)$ such that no proper subset of X is a generator of $f(X)$ is called a *minimal generator*. A minimum cardinality generator is called a *minimum generator*.

Intuitively, a set X generates $f(X)$ in one step. The infection process may continue applying recursively f to the new set of generated vertices. The *convex hull* $f^\infty(X)$ of a set $X \subseteq V$ (with respect to f) is the set of all vertices that will eventually be generated when X is the set of initially infected vertices. More formally, for every set X and $i > 1$, let $f^1(X) = f(X), f^i(X) = f(f^{i-1}(X))$ and $f^\infty(X) = lim_{i \to \infty} f^i(X)$. A set X is called a *recursive generator* of $f^\infty(X)$. If $f^\infty(X) = V$, X is called a *recursive generator* (a.k.a. *hull set*) for G.

For a graph G and an interval function f, several natural questions can be asked. For instance:

(a) What is the minimum size of a generator for G, i.e., the minimum size of a set $X \subseteq V$ such that $f(X) = V$? This graph invariant is called the *interval number* of G in the convexity defined by the function f, denoted here by $in_f(G)$. We stress that the interval number has received different names in the literature which usually depends on the convexity function f, such as *geodetic number* Hernando et al. (2005a) and *monophonic number* Paluga and Canoy Jr (2007).

(b) What is the minimum size of a recursive generator for G, i.e., the minimum size of a set $X \subseteq V$ whose convex hull is V? This graph invariant is called the *hull number* Everett and Seidman (1985) of G (with respect to f), denoted here by $hn_f(G)$.

Many other graph-convexity parameters have been considered in the literature like *convexity number* Chartrand et al. (2002c), *Radon number* (e.g., Dourado et al. (2012)), *Carathéodory number* (e.g, Dourado et al. (2013)), *rank number* Ramos et al. (2014), etc. In this work we focus on the interval number.

Several interval functions have been proposed and studied. For instance:

Geodetic convexity Dourado et al. (2009); Araujo et al. (2013): two contaminated vertices infect any vertex on a shortest path between them. This is similar to the classic notion of convexity in an Euclidean space, if one sees the segment between two points as the shortest path between them. This function is denoted here by sp. That is, $sp(X, v) = X \cup \{v\}$ if and only if v belongs to a shortest path between two vertices of X. In Section 2.2, the corresponding hull number (resp., interval number) of a graph G is denoted by $hn_{sp}(G)$ (resp., $in_{sp}(G)$).

Monophonic convexity Jamison (1982); Dourado et al. (2010b): two contaminated vertices infect any vertex on an induced path between them. The function is denoted by m and $m(X, v) = X \cup \{v\}$ if and only if v belongs to an induced path between two vertices of X. In Section 2.2, the corresponding hull number (resp., interval number) of a graph G is denoted by $hn_m(G)$ (resp., $in_m(G)$).

P_3**-convexity** Varlet (1976); Centeno et al. (2010): a vertex is contaminated if it has two contaminated neighbors, i.e. $P_3(X, v) = X \cup \{v\}$ if and only if $|N(v) \cap X| \geq 2$. In Section 2.2, the corresponding hull number (resp., interval number) of a graph G is denoted by $hn_{P_3}(G)$ (resp., $in_{P_3}(G)$).

P_3^***-convexity** Araújo et al. (2013): a vertex is contaminated if it has two contaminated non-adjacent neighbors, i.e. $P_3^*(X, v) = X \cup \{v\}$ if and only if there exist $x, y \in X \cap N(v)$, $x \neq y$ and $\{x, y\} \notin E(G)$. In Section 2.2, the corresponding hull number (resp., interval number) of a graph G is denoted by $hn_{P_3^*}(G)$ (resp., $in_{P_3^*}(G)$).

Here we study a new graph convexity called *cycle convexity*, that has been defined as follows in Araujo et al. (2018):

Definition 1 (Cycle convexity)**.** In cycle convexity, a vertex is contaminated if is has two *distinct* neighbors in a same connected component of the graph induced by the infected vertices. The corresponding interval function is denoted by cc, and $cc(X, v) = X \cup \{v\}$ if and only if there is a path P of order at least 2 in the subgraph $G[X]$ induced by X, such that both endpoints of P are neighbors of v.

In Araujo et al. (2018), it is shown that the hull number in cycle convexity of a 4-regular planar graph provides an upper bound for the *tunnel number* of the associated knot or link embedded in the 3-dimensional sphere.[(i)]. A tight upper bound for $hn_{cc}(G)$ is provided, when G is a 4-regular planar graph and it is proven that it is NP-complete to determine whether $hn_{cc}(G) \leq k$, given a planar graph G and a positive integer k.

1.2 Our results

In this paper, we further study cycle convexity and we focus on the corresponding interval number.

We first present some exact values for restricted cases and several bounds for the interval number of a graph in Sections 2 and 3. Although some of these bounds imply complexity results, we focus on the complexity analysis of computing the interval number in cycle convexity in Sections 4, 5 and 6.

Section 2 is devoted to some general basic results. We first present some trivial facts concerning simple graph classes and we characterize when $in_{cc}(G) = 2$. Then, we prove some relationships between cycle convexity and other more studied graph convexities (*e.g.*, P_3-convexity), as well as with domination problems in graphs (*i.e.*, domination number and connected 2-domination number). Doing so, there are

[(i)] Formal definitions of tunnel, knot or link are out of the scope of this paper. We refer to Clark (1980) for the interested reader.

several bounds in the literature concerning the above parameters that can be extended to the interval number in the cycle convexity Chellali et al. (2012); Bujtás and Jaskó (2017). We also show how this can be used in order to derive inapproximability results for the computation of the interval number.

Section 3 first presents some general bounds to the interval number in cycle convexity. Then, we study the interval number in_{cc} of square grids. We show that, for any grid G_n with n vertices, $n/2 \leq in_{cc}(G_n) \leq n/2 + o(n)$. Furthermore, we explain how to use our bounds in this section in order to design *parameterized* algorithms for computing in_{cc} in apex-minor free graphs, and so, in planar graphs and bounded-genus graphs. This part is based on bidimensionality theory Fomin et al. (2011).

In Section 4, we show that deciding whether $in_{cc}(G) \leq k$ is NP-hard when k is part of the input, and G belongs to the class of split graphs, or bipartite graphs, or planar graphs with maximum degree 6. To the best of our knowledge the complexity of computing $in(G)$ for planar graphs is open for other related graph convexities. Furthermore, the results for split graphs and for bipartite graphs are even stronger because both reductions are FPT-reductions from the Dominating Set problem. Thus, determining $in_{cc}(G)$ for any G in these two classes of graphs is $W[2]$-hard. Moreover, our results imply that determining $in_{cc}(G)$ for any G cannot be approximated in polynomial-time up to a factor $\mathcal{O}(\log n)$ in the classes of bipartite or split graphs.

On the positive side, in Section 5, we present polynomial-time algorithms to compute $in_{cc}(G)$ on cobipartite graphs and interval graphs. We emphasize our positive result for interval graphs, that is surprisingly intricate.

Finally, Section 6 is devoted to parameterized algorithms. Since we prove that computing $in_{cc}(G)$ is W[2]-hard when the parameter is the size of the solution, we investigate the behaviour of the problem when we consider other parameters. We present FPT algorithms to compute $in_{cc}(G)$ for $(q, q-4)$-graphs, where q is the parameter, for bounded treewidth graphs, and for graphs of bounded neighborhood diversity. The result for bounded treewidth graphs allows us to prove that $in_{cc}(G)$ can be computed in polynomial-time in the class of outerplanar graphs G. Our motivation to study these parameters comes from similar results obtained in different graph convexities Campos et al. (2015); Araujo et al. (2013); Araújo et al. (2013); Araujo et al. (2016).

Most of our results clearly show a strong relationship between the interval number for cycle convexity and the Minimum Dominating Set problem. It would be interesting to know whether the complexity of these two problems differ in some graph classes.

2 Preliminaries

For basic notions in graph theory, convexity in Euclidean spaces, computational complexity and parameterized complexity, the reader is referred to Bondy and Murty (2008); Flum and Grohe (2006); Rockafellar (1970); Garey and Johnson (1979).

All graphs considered in this paper are simple (i.e., without parallel edges nor loops), finite and undirected. Unless explicitly stated, graphs considered in this paper are also connected.

Throughout the paper, we use the following notations. Let $G = (V, E)$ be a graph. For any $v \in V$, $N(v) \subseteq V$ denotes the set of neighbors of v, and $N[v] = \{v\} \cup N(v)$ is the closed neighborhood of v. Let $E_v \subseteq E$ denote the set of edges incident to v. For any $X \subseteq V$, $N(X) = \bigcup_{v \in X} N(x) \setminus X$, i.e. it is the set of neighbors of the vertices in X that are out of X, and $N[X] = N(X) \cup X$. $G[X]$ is the subgraph induced by X in G. A vertex $v \in V$ is *universal* if $N[v] = V$.

In this paper, we consider *cycle convexity*, where a vertex gets contaminated if it is adjacent to two *distinct* vertices of a same contaminated connected component. More formally, for any $X \subset V$ and $v \in V \setminus X$, let $cc(X,v) = \{v\} \cup X$ if and only if there is a connected component A of $G[X]$ such that $|A \cap N(v)| \geq 2$. Otherwise, $cc(X,v) = X$. This means that the vertex v is generated by X if and only if there is a cycle C in G containing v and such that $V(C) \subseteq X \cup \{v\}$ (note that such a cycle is not necessarily unique). We say that this cycle C generates v for the set X.

Let us put $cc(X) = \bigcup_{v \in V \setminus X} cc(X,v)$. The interval number $in_{cc}(G)$ of G (in cycle convexity) is the smallest integer k such that there exists $X \subseteq V$ with $|X| \leq k$ and $cc(X) = V$. Finally, the hull number $hn_{cc}(G)$ of G (with respect to cycle convexity) is the smallest integer k such that there exists $X \subseteq V$, $|X| \leq k$ and $cc^{\infty}(X) = lim_{i \to \infty} cc^i(X) = V$.

2.1 Basic properties

Let us start with some simple observations that are used all along the paper.

The following two lemmas are straightforward.

Lemma 2. *Let $G = (V,E)$ be a simple graph and $X \subseteq V$. $cc(X) = V$ if and only if, for every $v \in V \setminus X$, there exist $x, y \in X \cap N(v)$, $x \neq y$, and x, y belong to a same connected component of $G[X]$.*

Proof: If $cc(X) = V$ then, for every $v \in V \setminus X$, there is a cycle C such that $V(C) \subseteq X \cup \{v\}$. In particular the two neighbours x, y of v in C are connected in $G[X]$. Conversely, suppose that, for every $v \in V \setminus X$, there exist $x, y \in X \cap N(v)$, $x \neq y$, and x, y belong to a same connected component of $G[X]$. The union of v with any induced xy-path in $G[X]$ is a cycle of which v is the only vertex not in X, thereby implying $cc(X,v) = X \cup \{v\}$. Therefore, $cc(X) = V$. □

Lemma 3. *For any n-node forest T, $in_{cc}(T) = n$.*
For any n-vertex cycle C_n, $n \geq 3$, $in_{cc}(C_n) = n - 1$.
For any complete n-vertex graph K_n, $n \geq 2$, $in_{cc}(K_n) = 2$.

Proof: The first statement comes from Lemma 2 and the fact that, in a tree, every vertex with degree at least 2 is a cut-vertex. The second statement also follows from Lemma 2 since, if some vertex $v \in C_n$ does not belong to a generator X, its two neighbors must belong to a same component of X and so $X = V(C_n) \setminus \{v\}$. Finally, again by Lemma 2, $in_{cc}(G) \geq 2$ for every graph with at least two vertices, and it is easy to check that $in_{cc}(K_n) \leq 2$. □

Theorem 4. *For any simple n-vertex m-edge connected graph G, $in_{cc}(G) = 2$ if and only if G has two distinct universal vertices. It can be decided in $\mathcal{O}(n+m)$-time.*

Proof: First, let us assume that $G = (V,E)$ admits a generator $X = \{u,v\} \subseteq V$ (with $u \neq v$), i.e. $cc(X) = V$. If $|V| = 2$, the result follows since G is connected. Let us assume that $|V| > 2$. By Lemma 2, for any $w \in cc(X) \setminus X$, w has two neighbors in a same contaminated connected component. This directly implies that $\{u,v\} \subseteq N(w)$ and $\{u,v\} \in E$. Hence, u and v are universal.

Reciprocally, if G has two (distinct) universal vertices u and v, then $X = \{u,v\}$ is clearly a generator for G, since any vertex of $V \setminus X$ is in a triangle with u and v. Hence, $in_{cc}(G) \leq 2$ and the equality follows since $|V| > 1$.

Note that the existence of two universal vertices can be checked in $\mathcal{O}(n+m)$-time, simply by computing the degree distribution. □

2.2 Relationship with other graph convexities

First, let us compare the hull number and interval number in cycle convexity with the corresponding invariants in geodetic and monophonic convexity.

In what follows, we say that two parameters are not comparable if none of the two can be upper-bounded by a function of the other for all graphs. By extension, we say that two convexities are not comparable if their respective hull number and interval number are not comparable.

Lemma 5. *Cycle convexity is not comparable to geodetic convexity nor to monophonic convexity.*

Proof: Let C_n be an n-vertex cycle. It is easy to check that: $hn_m(C_n) = in_m(C_n) = 2$ and $hn_{sp}(C_n) = in_{sp}(C_n) = 2$ if n is even, and $hn_{sp}(C_n) = in_{sp}(C_n) = 3$ otherwise. For any n-vertex complete graph K_n, $hn_{sp}(K_n) = hn_m(K_n) = in_{sp}(K_n) = in_m(K_n) = n$. Combining the above equalities with Lemma 3, we get that interval and hull numbers in cycle-convexity and geodetic (resp., monophonic) convexity are not comparable. □

Then, let us compare the hull number and interval number in cycle convexity with the corresponding invariants in P_3 (resp., P_3^*)-convexity.

Lemma 6. *In any graph G,*

$$hn_{P_3}(G) \leq hn_{cc}(G) \leq 2 \cdot hn_{P_3}(G) - 1 \leq 2 \cdot hn_{P_3^*}(G) - 1, \text{ and}$$

$$in_{P_3}(G) \leq in_{cc}(G) \leq 2 \cdot in_{P_3}(G) - 1 \leq 2 \cdot in_{P_3^*}(G) - 1.$$

Proof: For any $X \subseteq V(G)$, let $P_3(X)$ be the set of vertices obtained by adding to X all vertices that have at least two neighbors in X. That is, $P_3(X)$ is the set of vertices generated by X in P_3-convexity. For any $i \geq 2$, let $P_3^i(X) = P_3^{i-1}(P_3(X))$ and $P_3^1(X) = P_3(X)$.

It directly follows from Lemma 2 and the definition of P_3-convexity that $in_{P_3}(G) \leq in_{cc}(G)$ for any graph G. By a similar argument, it is easy to show that, for any $i \geq 1$ and for any $X \subseteq V(G)$, $cc^i(X) \subseteq P_3^i(X)$. Hence, $hn_{P_3}(G) \leq hn_{cc}(G)$ for any graph G.

Note that $hn_{P_3}(G) \leq hn_{P_3^*}(G)$ and $in_{P_3}(G) \leq in_{P_3^*}(G)$ for any graph G. Hence, it only remains to prove the upper bounds for cycle-convexity.

Let $X \subseteq V(G)$ be any recursive generator for G in P_3-convexity, i.e., $P_3^{\infty}(X) = lim_{i \to \infty} P_3^i(X) = V(G)$. Let $c \geq 1$ be the number of connected components of $G[X]$. We prove by induction on c that there exists $Y \subseteq V(G) \setminus X$, $|Y| \leq c - 1$ and $Y \cup X$ recursively generates G in cycle-convexity. This implies that $hn_{cc}(G) \leq |X| + c - 1 \leq 2|X| - 1$. Since it holds for any recursive generator X in P_3-convexity, this will imply that $hn_{cc}(G) \leq 2 \cdot hn_{P_3}(G) - 1$.

First, let us assume that $c = 1$. We prove by induction on i that, for any $i \geq 1$, $P_3^i(X) = cc^i(X)$. Indeed, at each step, $P_3^i(X)$ induces a connected subgraph of G. Moreover, any vertex v that gets generated (in P_3-convexity) in $P_3^i(X)$ has at least two neighbors in $P_3^{i-1}(X)$, thus, by Lemma 2, we also have $v \in cc(P_3^{i-1}(X))$. Hence, $P_3^i(X) \subseteq cc^i(X)$. By similar arguments, $cc^i(X) \subseteq P_3^i(X)$. Hence, if $c = 1$, then $hn_{cc}(G) \leq hn_{P_3}(G)$.

Now, let us assume that $G[X]$ has $c > 1$ connected components. Let us assume that X does not recursively generate G in cycle-convexity and let $y \in V(G) \setminus X$ be one of the first vertices that are not generated. More precisely, let i be the minimum integer such that $P_3^i(X) \setminus cc^i(X) \neq \emptyset$ (i exists since X recursively generates $V(G)$ in P_3-convexity) and let $y \in P_3^i(X) \setminus cc^i(X)$. Because y is generated by $P_3^{i-1}(X)$ in P_3-convexity, but not in cycle-convexity, it must have two neighbors in two distinct contaminated connected components of $G[P_3^{i-1}(X)]$. Note also that $G[P_3^{i-1}(X)]$ has at most c connected components, since new components cannot be created during the infection process. Hence, $P_3^{i-1}(X) \cup \{y\}$ recursively generates G in P_3-convexity and has at most $c-1$ connected components. By the induction hypothesis, there is $Y \subseteq V(G) \setminus (P_3^{i-1}(X) \cup \{y\})$ and $|Y| \leq c-2$ such that $Y \cup (P_3^{i-1}(X) \cup \{y\})$ recursively generates G in cycle-convexity. Finally, since $P_3^{i-1}(X) = cc^{i-1}(X)$ (by definition of i and by similar argument as in paragraph above), we have that $X \cup Y \cup \{y\}$ recursively generates G in cycle convexity and $|Y \cup \{y\}| \leq c-1$.

The proof of the fact that $in_{cc}(G) \leq 2 \cdot in_{P_3}(G) - 1$ is similar. □

Notice that the bounds of Lemma 6 are sharp, since on the one hand we have $in_{cc}(K_n) = in_{P_3}(K_n) = 2$ for any n-vertex clique K_n with $n \geq 2$, while on the other hand we have $in_{cc}(P_{2n+1}) = 2n+1$ and $in_{P_3}(P_{2n+1}) = in_{P_3^*}(P_{2n+1}) = n+1$ for any $(2n+1)$-node path P_{2n+1} (the same relationships hold for hull number). Furthermore, we stress that the interval number in P_3-convexity is exactly the 2-domination number of the graph. The *connected* 2-domination number is a natural upper-bound on the interval number in both the P_3-convexity and the cycle convexity. We prove with Lemma 6 that the connected 2-domination number of a given graph G has cardinality at most $2 \cdot in_{P_3}(G) - 1$.

By the previous lemma, combined with all inapproximability results on P_3-convexity Coelho et al. (2014), we get the following corollary. Namely, in bipartite graphs, computing the interval number for P_3-convexity cannot be approximated in polynomial-time up to an approximation ratio $\mathcal{O}(\log n)$ unless P=NP Coelho et al. (2014).

Theorem 7. *The parameter in_{cc} cannot be approximated up to a factor $\mathcal{O}(\log n)$ in polynomial-time (unless P=NP) in the class of bipartite graphs.*

In the remaining part of the paper, we only consider cycle convexity.

3 General bounds and Grids

This section is devoted to general lower bounds on the interval number of a graph (in cycle convexity), depending on its number of vertices and maximum degree. These lower bounds are then improved in the case of grids.

3.1 General lower bounds

Let X be a generator for a graph G. For any $v \in V \setminus X$, let $g_X(v)$ be a set consisting of two arbitrary distinct vertices in $N(v) \cap X$ that are in a same connected component of $G[X]$ (two such vertices must always exist by Lemma 2). When X is clear from the context, we write $g(v)$ instead of $g_X(v)$. Note that the two vertices in $g(v)$ are the ends of a path in X which generates the vertex v.

Theorem 8. *For any n-vertex graph G with maximum degree Δ, $in_{cc}(G) \geq \frac{2n}{\Delta+1}$.*

Proof: Let X be a generator for G. Let $x \in X$ be an isolated vertex in $G[X]$ (i.e., x has no neighbors in X). Then, for any $v \in V \setminus X$, $x \notin g(v)$ since x cannot generate any vertex. For any vertex $x \in X$ with at least one neighbor in X, there are at most $\Delta - 1$ vertices v in $V \setminus X$ such that $x \in g(v)$ (since x has at most $\Delta - 1$ neighbors not in X).

Hence, for any vertex $x \in X$, there are at most $\Delta - 1$ vertices v in $V \setminus X$ such that $x \in g(v)$. That is, $\bigcup_{v \in V \setminus X} g(v) \subseteq X$ and any $x \in X$ appears in at most $\Delta - 1$ sets $g(v)$. Moreover, recall that $|g(v)| = 2$ for any $v \notin X$.

Therefore, $2|V \setminus X| = 2(n - |X|) = \sum_{v \in V \setminus X} |g(v)| \leq (\Delta - 1)|X|$. Hence, $|X| \geq \frac{2n}{\Delta+1}$. $\square$

In the next theorem, the above bound is refined using the structure of the subgraph induced by the generator. It is easy to see that this bound is tight by considering, for instance, any cycle.

Theorem 9. *For any n-vertex graph G with maximum degree Δ, and for any generator X for G such that $G[X]$ has $r \geq 1$ connected components: $|X| \geq \frac{2(n-r)}{\Delta}$.*

Proof: Let $x \in X$ and let $deg_X(x) \geq 0$ denote its degree in $G[X]$. For any $0 \leq i \leq \Delta$, let X_i be the subset of vertices of X with degree i in $G[X]$, i.e. $X_i = \{x \in X \mid deg_X(x) = i\}$. As in the proof of Theorem 8, if $x \in X_0$, then x cannot appear in any set $g(v)$, $v \in V \setminus X$. If $x \in X_i$ for $i > 0$, then x has at most $\Delta - i$ neighbors in $V \setminus X$ and, therefore, it appears in at most $\Delta - i$ sets $g(v)$, $v \in V \setminus X$.

Hence, we get that $2(n - |X|) = \sum_{v \in V \setminus X} |g(v)| \leq \sum_{1 \leq i \leq \Delta} (\Delta - i)|X_i|$.

Let C be any connected component of $G[X]$ and let $X_i^C = X_i \cap C$. Moreover, let $\mathcal{C}_X$ be the set of connected components of $G[X]$ that are not reduced to a single vertex. The previous inequality can be rewritten as:

$$\sum_{v \in V \setminus X} |g(v)| \leq \sum_{C \in \mathcal{C}_X} \Big(\sum_{1 \leq i \leq \Delta} (\Delta - i)|X_i^C| \Big). \tag{1}$$

Let us prove that, for any $C \in \mathcal{C}_X$,

$$\sum_{1 \leq i \leq \Delta} (\Delta - i)|X_i^C| \leq 2(\Delta - 1) + (\Delta - 2)(|C| - 2). \tag{2}$$

In order to prove Inequality 2, let us prove more generally that for every subset $C' \subseteq C$ inducing a connected subgraph in $G[C]$.

$$\sum_{1 \leq i \leq \Delta} (\Delta - i)|X_i^{C'}| \leq 2(\Delta - 1) + (\Delta - 2)(|C'| - 2),$$

where $X_i^{C'}$ denotes the vertices of C' with degree i in $G[C']$ (i.e., with i neighbors in C'). The proof is by induction on $|X_1^{C'}|$. If there are at most 2 vertices with degree 1 in C', i.e. $|X_1^{C'}| \leq 2$, then the result holds since $\sum_{1 \leq i \leq \Delta} (\Delta - i)|X_i^{C'}| = (\Delta - 1)|X_1^{C'}| + \sum_{2 \leq i \leq \Delta} (\Delta - i)|X_i^{C'}| \leq (\Delta - 1)|X_1^{C'}| + (\Delta - 2) \sum_{2 \leq i \leq \Delta} |X_i^{C'}|$, and $\sum_{2 \leq i \leq \Delta} |X_i^{C'}| = |C'| - |X_1^{C'}|$.

Else, $|X_1^{C'}| \geq 3$, let us assume that the induction hypothesis holds true for every subset $C'' \subseteq C$ inducing a connected subgraph of $G[C]$ and such that $|X_1^{C''}| < |X_1^{C'}|$. Let $v_1 \in X_1^{C'}$ have degree 1 in $G[C']$. Let v_m be a closest vertex to v_1 in $G[C']$ that has degree at least three in $G[C']$. Such a vertex exists since $G[C']$ connected, and $|X_1^{C'}| \geq 3$ implies that C' is not reduced to a path. Furthermore, let $P = (v_1, \cdots, v_m)$, $m \geq 2$, be a shortest $v_1 v_m$-path of $G[C']$. By the choice of v_m, all internal vertices of P have degree 2 in C'. Let $C'' = C' \setminus \{v_1, \cdots, v_{m-1}\}$ (note that C'' induces a connected subgraph) and let $d_m = deg_{C'}(v_m) - 1 \geq 2$ be the degree of v_m in $G[C'']$. Since there is one vertex of degree 1 less in $G[C'']$ than in $G[C']$, i.e. $|X_1^{C''}| = |X_1^{C'}| - 1$, we may apply the induction hypothesis on C'' and $\sum_{1 \leq i \leq \Delta} (\Delta - i)|X_i^{C''}| \leq 2(\Delta - 1) + (\Delta - 2)(|C''| - 2)$.

Moreover, $\sum_{1\leq i\leq\Delta}(\Delta-i)|X_i^{C'}| = (\Delta-1)+(\Delta-2)(m-2)-1+\sum_{1\leq i\leq\Delta}(\Delta-i)|X_i^{C''}|$. In the last sum, the first term $\Delta-1$ comes from v_1, the term $(\Delta-2)(m-2)$ comes from $v_2,\cdots,v_{m-1}$ and the -1 comes from the fact that v_m has degree one more in $G[C']$ than in $G[C'']$.

Altogether, we get that:

$$\sum_{1\leq i\leq\Delta}(\Delta-i)|X_i^{C'}| \leq (\Delta-1)+(\Delta-2)(m-2)-1+2(\Delta-1)+(\Delta-2)(|C''|-2).$$

And, since $(\Delta-1)+(\Delta-2)(m-2)-1+2(\Delta-1)+(\Delta-2)(|C''|-2) = 2(\Delta-1)+(\Delta-2)(|C'|-2)$, we get:

$$\sum_{1\leq i\leq\Delta}(\Delta-i)|X_i^{C'}| \leq= 2(\Delta-1)+(\Delta-2)(|C'|-2).$$

This concludes the proof of Inequality 2.

Let $Y\subseteq X$ be the set of isolated vertices in X (i.e., the vertices in X that have no neighbors in X). Let $r'=|Y|$ and let r'' be the number of connected components of $X\setminus Y$. Let $r=r'+r''$ be the number of connected components of X. From Inequalities 1 and 2, we get that

$$2(n-|X|) = \sum_{v\in V\setminus X}|g(v)| \leq 2r''(\Delta-1)+(\Delta-2)(|X\setminus Y|-2r'').$$

Note that $Z = 2r'(\Delta-1)+(\Delta-2)(|Y|-2r')\geq 0$ (since $|Y|=r'$).

Therefore,

$$2(n-|X|) = \sum_{v\in V\setminus X}|g(v)| \leq 2r''(\Delta-1)+(\Delta-2)(|X\setminus Y|-2r'')+Z.$$

Since $2r''(\Delta-1)+(\Delta-2)(|X\setminus Y|-2r'')+Z = 2(r'+r'')(\Delta-1)+(\Delta-2)(|X\setminus Y|+|Y|-2r''-2r') = 2r(\Delta-1)+(\Delta-2)(|X|-2r)$, we get that

$$2(n-|X|) = \sum_{v\in V\setminus X}|g(v)| \leq 2r(\Delta-1)+(\Delta-2)(|X|-2r).$$

Hence, $\frac{2(n-r)}{\Delta}\leq|X|$. □

3.2 Application to the grids

In the proof of Theorem 9, the Inequality 2 can be refined in some cases. In particular, we now show that it is the case for any generator of a grid.

Corollary 10. *For any grid G with n vertices, $in_{cc}(G)\geq n/2$.*

Proof: Notations are the same as for Theorem 9. Let X be a fixed generator for G of minimum size. Consider Inequality 1. For the upper-bound, we assume the worst-case when (i) all vertices in X have degree $\Delta=4$, and (ii) for all $x\in X$, $x\in g(v)$ for every of its neighbors v that is not in X. So, in the particular case when there is a vertex in X that has degree strictly less than Δ, the upper-bound can be decreased by 1. Similarly, in the particular case when there exists $x\in X$ such that $x\notin g(v)$ for some of its neighbors $v\notin X$, the upper-bound can also be decreased by 1.

We now claim that for every (non isolated) component C of X, we fall in one of the two above cases: precisely, there exist two vertices in C that have degree strictly less than Δ or that have a neighbor that they do not generate. Indeed, the leftmost (resp., rightmost, topmost, downmost) vertex of C has either the first property (if it is at the border of the grid) or the second one (it cannot generate the vertex that is left, resp., right, top, down to it). As a result, the upper-bound of Inequality 1 can be decreased by $2r$, with r being the number of components in X. Combined with Inequality 2, one obtains:

$$2(n-|X|) = \sum_{v\in V\setminus X} |g(v)| \leq 2r(\Delta-1) + (\Delta-2)(|X|-2r) - 2r = (\Delta-2)|X|.$$

Hence, $in_{cc}(G) = |X| \geq \frac{2n}{\Delta} = n/2$. □

Since the set that consists of one row plus one column every two columns (plus, if the number of columns is even, one vertex every three rows for the last column) is a generator, we get that

Theorem 11. *For any square grid G with n vertices, $n/2 \leq in_{cc}(G) = n/2 + O(\sqrt{n})$.*

To conclude this section, let us give some algorithmic applications of our result for grids, using techniques from the bidimensionality theory Fomin et al. (2011). First, let us generalize it in a larger graph class.

Let us remind that a planar triangulation of a planar graph G is a planar supergraph of G whose faces are bounded by triangles. A *partially triangulated* $(r \times r)$*-grid* is any graph that contains an $(r \times r)$-grid as a subgraph and is a subgraph of some planar triangulation of the same $(r \times r)$-grid.

The graph Γ_k Fomin et al. (2011) is obtained from a $(k \times k)$-grid by triangulating its internal faces such that all internal vertices become of degree 6, all non-corner external vertices are of degree 4, and then one corner of degree two is joined by edges with all vertices of the external face.

Lemma 12. *For any $k \in \mathbb{N}$, $in_{cc}(\Gamma_k) = \Omega(k^2)$.*

The proof of the above lemma is similar to the one of Theorem 8, by noticing that all vertices have bounded degree at most $\Delta = 6$ except the corner adjacent to the external face. This latter vertex has degree $\Theta(k)$, hence its contribution is negligible in the equations of the proof of Theorem 8. Precisely, it becomes $2|V \setminus X| = 2(k^2 - |X|) = \sum_{v\in V\setminus X} |g(v)| \leq (\Delta-1)(|X|-1) + \Theta(k)$. Hence, $|X| \geq \Omega(k^2)$.

Let $tw(G)$ be the treewidth of a graph (defined in Section 6.1). On the algorithmic point of view, it is known that for any graph parameter Π stable under edge-contraction, if $\Pi(\Gamma_k) = \Omega(k^2)$ and $\Pi(G)$ can be computed in $f(tw(G)) \cdot n^{\mathcal{O}(1)}$-time for some computable function f, then deciding $\Pi(G) \leq k$ can be done in $f(\mathcal{O}(\sqrt{k})) \cdot n^{\mathcal{O}(1)}$-time for any *apex-minor-free graph G* (see for instance (Fomin et al., 2011, Theorem 4)). Notice that in_{cc} is trivially stable under edge-contractions. Furthermore, we will prove in Theorem 26 that $in_{cc}(G)$ can be computed in $2^{\mathcal{O}(tw(G)\log(tw(G))} \cdot n$-time. Then, altogether combined with Lemma 12, we obtain the following result:

Theorem 13. *Deciding $in_{cc}(G) \leq k$ can be done in $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n$-time for any apex-minor-free graph G.*

4 Hardness for restricted cases

This section is devoted to the time-complexity of computing $in_{cc}(G)$ in various graph classes. Clearly, the problem is in NP. The reductions presented below take advantage of the relationship between Minimum Dominating Set Problem and the problem of computing the interval number for cycle convexity.

4.1 Relationship with Minimum Dominating Set and case of Split graphs

Given a graph $G = (V, E)$, a set $D \subseteq V$ is a *dominating set* of G if $N[D] = V$. Let $\gamma(G)$ denote the minimum size of a dominating set of G. Let G^* be the graph obtained from G by adding a universal vertex u, i.e., $V(G^*) = V \cup \{u\}$ and $E(G^*) = E \cup \{\{v, u\} \mid v \in V\}$.

Theorem 14. *For any graph G, $in_{cc}(G^*) = \gamma(G) + 1$.*

Proof: Let u be the universal vertex added to obtain G^*.

Let us first prove $in_{cc}(G^*) \leq \gamma(G) + 1$. Let D be a dominating set of G. We claim that $D \cup \{u\}$ is a generator for G^*. Indeed, for any $v \notin D \cup \{u\}$, there is a vertex $w \in D$ that dominates v. Hence, v is in the triangle $\{v, w, u\}$ and it is generated by $D \cup \{u\}$ in G^*.

For the opposite direction, let X be a generator for G^*. If $u \notin X$, let C be any cycle that generates u for X and let $v \in X$ be any vertex of C. We claim that $X' = X \cup \{u\} \setminus \{v\}$ is a generator for G^*. Clearly, C generates v for X'. Moreover, for any vertex w in $V(G) \setminus X$, there is a cycle C_w that generates w for X. If $v \notin C_w$, then w is still generated by C_w for X'. Otherwise, let C'_w be the cycle obtained from C_w by replacing v by u (C'_w exists since u is universal). Then, C'_w generates w for X'. Hence, we may assume that X contains u. Because X is a generator for G^* containing u, any vertex in $V(G) \setminus X$ has at least one neighbor in $X \setminus \{u\}$ (by Lemma 2). Hence, $X \setminus \{u\}$ is a dominating set of G. □

Notice that Theorem 14 implies that, if the Minimum Dominating Set problem is NP-hard, or even not approximable, in some graph class $\mathcal{G}$ closed under the addition of a universal vertex, then it also holds for in_{cc}.

A graph is a *split graph* if its vertex set can be partitioned into two sets, inducing a clique and an independent set respectively. Computing γ in split graphs is NP-hard and cannot be approximated up to a factor $\mathcal{O}(\log n)$ unless P=NP Chlebík and Chlebíková (2008); Dinur and Steurer (2014).

Corollary 15. *Computing in_{cc} in split graphs is NP-hard and cannot be approximated up to a factor $\mathcal{O}(\log n)$ unless P=NP.*

4.2 Bipartite graphs

We consider the MINIMUM DOMINATING SET Problem (MDS) which takes a graph $G = (V, E)$ and an integer k as inputs and asks whether there exists a set $X \subseteq V$, $|X| \leq k$ and $N[X] = V$. It is well known that this problem is NP-hard, and that:

Theorem 16. *Flum and Grohe (2006) Minimum Dominating Set Problem is NP-hard and W[2]-hard in bipartite graphs.*

Theorem 17. *Computing in_{cc} is NP-hard and W[2]-hard in bipartite graphs.*

Proof: To prove the theorem, we propose an FPT-reduction from MDS (i.e., a reduction preserving both the size of the problem and the parameter).

Let $G = (A \cup B, E)$ and $k \in \mathbb{N}$ be a bipartite instance of MDS.

Let G^* be obtained from G by adding a vertex a adjacent to all vertices of A, a vertex b adjacent to all vertices in B and the edge $\{a, b\}$. Clearly, G^* is bipartite. We claim that there is a dominating set of size k in G if and only if $in_{cc}(G^*) \leq k + 2$. Combined with Theorem 16, this claim will prove the result.

Let X be any dominating set of G with size k. Let $X^* = X \cup \{a, b\}$. It is easy to check that $cc(X^*) = V(G^*)$ and therefore, $in_{cc}(G^*) \leq k + 2$.

On the other hand, let X^* be a minimum generator for G^* (i.e., $cc(X^*) = V(G^*)$) such that $|X^*| \leq k+2$. If $a, b \in X^*$, then it is easy to check that $X^* \setminus \{a,b\}$ dominates $V(G)$. Indeed, suppose that $v \in V(G) \setminus X^*$ (w.l.o.g., $v \in A$) and $N(v) \cap X^* \subseteq \{a,b\}$. Then, by construction of G^*, $N(v) \cap X^* = \{a\}$ and X^* cannot generate v by Lemma 2.

It remains to prove that, if there exists a set of size $k+2$ that generates $V(G^*)$, then there is such a set that contains a and b. Indeed, assume that $a \notin X^*$. Therefore, there is a cycle C in G^* where a is the single vertex not in X^*. In particular, there is a vertex $c \in V(C) \cap X^* \cap A$. We show that $X' = X^* \cup \{a\} \setminus \{c\}$ is a generator for G^*. Indeed, c is clearly generated by C in X'. Moreover, for any cycle C' containing c that generates a vertex v for X^* (notice that C' cannot contain a), the cycle C'' obtained by replacing c with a in C' (it can be done since $N(c) \subset A$ and $A \subset N(a)$) generates v for X'. Hence, X' will generate the same vertices as X^*. □

4.3 Planar graphs with bounded degree

Finally, let us consider the case of planar graphs with bounded degree. Let us observe that by Theorem 13, the problem is FPT in planar graphs when the parameter is the size of the solution.

Theorem 18. *The problem of computing $in_{cc}(G)$ in planar graphs G with degree at most 6 is NP-hard.*

Proof:

Let $G = (V, E)$ be a graph with $n = |V|$ vertices and $m = |E|$ edges. The graph G' is obtained from G by adding $3m$ new vertices and corresponding edges as follows. For every $e \in E$, we add a triangle (x_e, y_e, z_e), then we make the vertex x_e adjacent to both vertices incident to e. Observe that if G has maximum degree Δ, then G' has maximum degree at most $\max\{2\Delta, 4\}$, and furthermore, if G is planar, then so is G'. Let us now prove that $in_{cc}(G') = \gamma(G) + 2m$, where $\gamma(G)$ is the domination number of G. Since it is NP-hard to compute the domination number of planar graphs with maximum degree three Garey and Johnson (1979), the latter will prove that it is NP-hard to compute $in_{cc}(G)$ even when G is a planar graph of maximum degree six.

Let us prove first that $in_{cc}(G') \leq \gamma(G) + 2m$. Consider a minimum-size dominating set D of G. The set S is constructed from D by adding the two vertices x_e, y_e for every $e \in E$. We claim that S is a generator for G'. Indeed, since for every $e \in E$, vertex z_e has its two adjacent neighbors x_e, y_e in S, the $3m$ vertices $x_e, y_e, z_e,\ e \in E$ are all generated by S. Furthermore, for every vertex $v \in V \setminus D$, since D is a dominating set there is $u \in D$ that is adjacent to vertex v. Let $e = \{u, v\} \in E$. The two adjacent vertices x_e, u are neighbors of vertex v in S, hence v is generated by S. As a result, $in_{cc}(G') \leq |S| = 2m + \gamma(G)$.

Conversely, let us prove that $in_{cc}(G') \geq \gamma(G) + 2m$. Let S be a minimum-size generator for G'. For every $e \in E$, if $y_e \notin S$, then since S is a generator for G', the only two neighbors x_e, z_e of vertex y_e must be in S. The same holds for vertex z_e. As a result, for every $e \in E$, there must be at least two of x_e, y_e, z_e in S. Now, let $D = S \cap V$. In order to prove the lower-bound, it suffices to prove that D is a dominating set of G. By contradiction, suppose the existence of a vertex $v \in V \setminus D$ with no neighbor in D. Since S is a generator for G', by Lemma 2, vertex v has two neighbors in some connected component of $G[S]$. Furthermore, since we assume that v has no neighbor in $D = V \cap S$, there must be $x_e, x_{e'} \in S$ such that there is an $x_e x_{e'}$-path in $G'[S]$ and $e, e' \in E$ are two edges incident to vertex v. Let $e = \{u, v\}, e' = \{u', v\}$. By construction of G', any $x_e x_{e'}$-path in S must pass by the two vertices u, u', thereby contradicting that v has no neighbors in $V \cap S$. As a result, D is a dominating set of V, and so, $in_{cc}(G') = |S| \geq |D| + 2m \geq \gamma(G) + 2m$. □

5 Polynomial-time algorithms

This section is devoted to polynomial-time algorithms to compute the interval number in several graph classes.

5.1 Cobipartite graphs

Recall that a graph is *cobipartite* if it is the complement of a bipartite graph. In other words, its vertex set can be partitioned into two cliques.

Theorem 19. *Let $G = (V, E)$ be a n-vertex cobipartite graph with $n > 1$. Then, $in_{cc}(G) \leq 4$, and:*

- *$in_{cc}(G) = 2$ if and only if it contains two universal vertices;*
- *otherwise, $in_{cc}(G) = 3$ if and only if one of the two following conditions hold:*
  - *either there exists $u \in V$ such that $in_{cc}(G \setminus u) = 2$;*
  - *or for any partition of V into two cliques A and B, one of them contains two vertices that dominate the other one.*

Moreover, $in_{cc}(G)$ can be computed in $\mathcal{O}(n^3)$-time.

Proof: Let $G = (A \cup B, E)$ where A and B induce disjoint cliques. Given a co-bipartite graph G, A and B can be computed in time $\mathcal{O}(n^2)$ by taking the complement of G, which is bipartite, then bicolouring this complement (in this situation, A and B can be chosen as the two colour classes for the complement).

Clearly, taking two vertices in each clique (or one vertex if a clique is reduced to a single vertex) generates the whole graph. Hence $in_{cc}(G) \leq 4$. Furthermore, we have $in_{cc}(G) > 1$, and by Theorem 4, $in_{cc}(G) = 2$ if and only if there are two universal vertices in G (which can be checked in time $\mathcal{O}(n^2)$). So, we are left to characterize when $in_{cc}(G) = 3$. In what follows, let us assume that $in_{cc}(G) \in \{3, 4\}$.

Suppose that $in_{cc}(G) \geq 3$. First, for any $u \in V$, by Theorem 4, it can be checked in time $\mathcal{O}(n^2)$ whether $in_{cc}(G \setminus u) = 2$. If such a vertex is found, then $in_{cc}(G) = 3$. The case is solved.

From now on, assume that $in_{cc}(G) \geq 3$ and, for every vertex $u \in A \cup B$, $in_{cc}(G \setminus u) > 2$. We will show that in this case $in_{cc}(G) = 3$ if and only if one of the cliques, A or B, contain two vertices that dominate the other clique.

Suppose $in_{cc}(G) = 3$, and let S be a generator of size three that contains vertices of both cliques. W.l.o.g, say that $|S \cap A| = 2$, and let $S \cap B = \{u\}$ and $S' = S \cap A$. Let us show that S' dominates B. Indeed, u must have some neighbor in S'. Otherwise, by Lemma 2, each vertex in $cc(S) \setminus S$ would have to be adjacent to both vertices in S', and S' would be a generator for $G \setminus u$ of size 2. This is a contradiction. On the other hand, since $S \cap B$ contains only one vertex, by Lemma 2, every vertex in $B \setminus S$ must be adjacent to at least one of the vertices in S'. Hence, S' dominates B and $|S'| = 2$.

Now, let S be a generator of size three that is included in one of the cliques, w.l.o.g., say $S \subseteq A$. By Lemma 2, all vertices in B are adjacent to at least two vertices in S. So we may pick any $v \in S$ and $S' = S \setminus \{v\}$ still dominates B and $|S'| = 2$.

For the opposite direction, w.l.o.g. suppose that there are $x, y \in A$ such that $B \subset N(\{x, y\})$. Pick any vertex $z \in B$ and let $S = \{x, y, z\}$. It is easy to check that S generates V and, by previous remarks, $in_{cc}(G) = 3$.

The condition for $in_{cc}(G) = 3$ can be verified in $\mathcal{O}(n^3)$-time by considering all the possible pairs x, y in A and B. $\square$

5.2 Interval graphs

This section is devoted to a dynamic programming algorithm to compute a minimum size generator for the cycle convexity in interval graphs.

An *interval* graph is the intersection graph of a set of segments of the line. In other words, each of its vertices corresponds to a segment in $\mathbb{R}$ and two vertices are adjacent if and only if the corresponding segments intersect Lekkeikerker and Boland (1962). A graph is *chordal* if it has no induced cycle of size at least 4 Blair and Peyton (1993). Interval graphs are chordal. In particular, it is well known that any interval graph G admits a clique-path, that is an ordering of its maximal cliques such that, for any vertex v of G, the maximal cliques containing v are contiguous in the ordering Blair and Peyton (1993).

This section is devoted to the proof of the following theorem.

Theorem 20. *If G is an interval graph, then $in_{cc}(G)$ and a minimum generator can be computed in $\mathcal{O}(n^3)$-time.*

Since the proof is technical we have divided it into several parts. First, we present some structural results on the generators for interval graphs, that will be the cornerstone for the algorithm of Theorem 20. These properties can be exploited in order to design a dynamic programming algorithm that computes $in_{cc}(G)$, in interval graphs, in exponential time (Lemma 25). Then, we explain how to reduce its time-complexity using the structure of interval graphs.

Notations. Let us introduce additional notations for the proof. Given an interval graph $G = (V, E)$, let $P = (X_1, X_2, \dots, X_l)$ be a clique-path of G, that is, the set of maximal cliques of G ordered in such a way that, for any vertex v of G, the maximal cliques containing v form an interval (subpath) P_v in P. By convention $X_0 = X_{l+1} = \emptyset$. Let us set $Y_i = \bigcup_{j=1}^{i} X_j,\ V_i = Y_i \setminus X_{i+1}$. Note that $V_i \setminus V_{i-1} = X_i \setminus X_{i+1}$.

The following properties are the cornerstone of our algorithm.

Recall that interval graphs are chordal. As a warm up, we prove a useful lemma on generators for chordal graphs.

Lemma 21. *For any generator S of a chordal graph G, every vertex $x \notin S$ has two adjacent neighbors in S.*

Proof: Since $x \notin S$, it is generated by a cycle C where it is the unique vertex not in S. Let C be the shortest among such cycles, and u, v be the neighbors of x on C. Notice that C must be an induced cycle, otherwise there would exist a shorter cycle C' that also generates x. Since G is a chordal graph, C must be a triangle, and therefore u and v are adjacent. That proves the lemma. □

We will often use the above lemma in what follows.

Lemma 22. *Let $G = (V, E)$ be an interval graph and $P = (X_1, X_2, \dots, X_l)$ be a clique-path of G. For any generator S for G, the vertices in V_i are generated by $S \cap Y_i$.*

Proof: Since G is an interval graph, and so, chordal, by Lemma 21 every vertex of $V \setminus S$ has two adjacent neighbors in S. In particular, V_i is generated by $S \cap N[V_i] = S \cap Y_i$. □

Next, we describe in Lemma 23 a procedure which given a generator S intersecting some ordered subset T, ensures the existence (under some conditions) of a generator S^* of the same size as S and containing the largest element in T. Applications to interval graphs are then discussed in the subsequent lemmas.

Lemma 23. *Let $G = (V, E)$ be an interval graph, $P = (X_1, X_2, \ldots, X_l)$ be a clique-path of G and S be a generator for G. Suppose that there exist vertices $v_i^1, \ldots, v_i^k \in S$ and $u_i^1, \ldots, u_i^k \notin S \cup V_{i-1}$ such that the following hold:*

- *the vertices of $Y_{i-1} \cup \{v_i^1, \ldots, v_i^k\}$ are generated by $S^* = (S \setminus \{v_i^1, \ldots, v_i^k\}) \cup \{u_i^1, \ldots, u_i^k\}$;*
- *and for every $1 \leq j \leq k$ $N[v_i^j] \setminus V_{i-1} \subseteq N[u_i^j] \setminus V_{i-1}$.*

Then, S^* is also a generator for G.

Proof: By the hypothesis the set S^* is a generator for $Y_{i-1} \cup \{v_i^1, \ldots, v_i^k\}$. Let us prove that every vertex $x \in V \setminus (Y_{i-1} \cup \{v_i^1, \ldots, v_i^k\})$ can be generated with S^*. If $x \in S^*$, then we are done. Otherwise, $x \notin S^*$, and so, $x \notin S$. By Lemma 21, since G is chordal, $x \notin S$ implies the existence of two adjacent neighbors of x in S, denoted by y and z. Note that $y, z \notin V_{i-1}$ since $x \notin Y_{i-1}$, and for every j, $u_i^j \notin \{y, z\}$ since $u_i^j \notin S$. Let us define $y^* = y$ if $y \in S^*$; otherwise, $y = v_i^j$ for some j and we define $y^* = u_i^j$. We define z^* in a similar fashion. By the hypothesis, $N[y] \setminus V_{i-1} \subseteq N[y^*] \setminus V_{i-1}$ and similarly $N[z] \setminus V_{i-1} \subseteq N[z^*] \setminus V_{i-1}$. Therefore, since $x \notin V_{i-1}$ and by construction $y^*, z^* \notin V_{i-1}$ are distinct, vertex x has two adjacent neighbors $y^*, z^* \in S^*$. Hence every vertex $x \in V \setminus (Y_{i-1} \cup \{v_i^1, \ldots, v_i^k\})$ can be generated with S^*. As a result, S^* is a generator for G. □

Our algorithm for Theorem 20 is based on a technical consequence of the above lemma. We describe it next.

Lemma 24. *Let $G = (V, E)$ be an interval graph, $P = (X_1, X_2, \ldots, X_l)$ be a clique-path of G and S be a generator for G. Furthermore, let $T \subseteq Y_i$ be such that $|T \cap S| \geq k$. Let $u_i^1, \ldots, u_i^k \in T$ be such that for every j, u_i^j maximizes $N[u_i^j] \setminus V_{i-1}$ in $T \setminus \{u_i^{j'} \mid j' < j\}$. If $(S \setminus T) \cup \{u_i^1, \ldots, u_i^k\}$ generates $Y_{i-1} \cup T$ and $u_i^1, \ldots, u_i^k \notin V_{i-1}$, then there exists a generator S^* for G such that $|S^*| \leq |S|$ and $u_i^1, \ldots, u_i^k \in S^*$.*

Proof: We assume w.l.o.g. (up to removing $S \cap \{u_i^1, \ldots, u_i^k\}$ from T) that $u_i^1, \ldots, u_i^k \notin S$. Let $v_i^1, \ldots v_i^k \in T \cap S$, that exist by the hypothesis. First note that since $T \subseteq Y_i$ and G is interval, the subsets in $\{N[v] \setminus V_{i-1} \mid v \in T\}$ can be ordered by inclusion. Therefore, let us assume w.l.o.g. that $N[v_i^j] \setminus V_{i-1} \subseteq N[v_i^{j+1}] \setminus V_{i-1}$ for any $j < k$, by the choices $u_i^1, \ldots, u_i^k$ we have $N[v_i^j] \setminus V_{i-1} \subseteq N[u_i^j] \setminus V_{i-1}$ for any j. Therefore, the result follows from Lemma 23 applied to S, $v_i^1, \ldots, v_i^k$ and $u_i^1, \ldots, u_i^k$. □

Equipped with the above lemmas, we now describe a first (exponential-time) algorithm for computing a minimum-size generator for interval graphs.

Lemma 25. *If G is an interval graph, then there exists a dynamic programming algorithm that computes $in_{cc}(G)$ and a minimum generator (in exponential-time).*

Proof: Let $P = (X_1, X_2, \ldots, X_l)$ be a clique-path of G. Note that P can be computed in $\mathcal{O}(n + m)$-time Blair and Peyton (1993). Our purpose is to compute $in_{cc}(G)$ by dynamic programming on the clique-path.

Iteratively, for $i = 1$ to l, the algorithm computes a set $\mathcal{S}_i$ of subsets of vertices such that, for any $S \in \mathcal{S}_i$, $S \subseteq Y_i$ and S generates V_i. Initially, $\mathcal{S}_1$ consists of all possible subsets of $Y_1 = X_1$ that generates V_1. Then, assuming that $\mathcal{S}_i$ has been built, the next set $\mathcal{S}_{i+1}$ is computed as follows. For any set $S \in \mathcal{S}_i$, the sub-procedure CONVEX(i, S) (described below) is applied. It returns one or two subsets, each of which are added to $\mathcal{S}_{i+1}$. Finally, the algorithm outputs the minimum size of a set in $\mathcal{S}_l$ and a corresponding set.

The correctness of the above algorithm will directly follows from the properties and the correctness of the sub-procedure CONVEX that we detail below. Moreover, as it will be clear, the sub-procedude CONVEX performs in polynomial time. Therefore, the exponential-time complexity of the algorithm only relies on the potentially exponential size of the sets $\mathcal{S}_i$. This drawback will be handled in the next lemma.

The sub-procedure CONVEX takes as input $i \in \{1, \dots, l\}$, and any subset $S_{i-1} \subseteq Y_{i-1}$ that is a generator for V_{i-1}. It returns one (or possibly two) superset(s) S_i of S_{i-1} that is (are) a generator(s) for V_i. We will show that at least one subset S_i that is output by the sub-procedure is contained in a generator for G that is of minimum size among all those containing S_{i-1}. In particular, this will prove that, if we start with $S_0 = \emptyset$, then the algorithm will compute in l steps a minimum-size generator for G.

We both describe the algorithm and prove its correctness simultaneously.

Let $i \leq l$ be fixed. Several cases and sub-cases have to be considered.

- if V_i is generated by S_{i-1}, then since $S_i = S_{i-1}$ generates V_i, we set CONVEX$(i, S_{i-1}) = \{S_i\} = \{S_{i-1}\}$ and it satisfies all the desired requirements for the sub-procedure;

- else, S_{i-1} does not generate $V_i \backslash V_{i-1} = X_i \backslash X_{i+1}$. Since X_i is a clique, it implies that $|S_{i-1} \cap X_i| \leq 1$. Indeed, otherwise there would be $u, v \in S_{i-1} \cap X_i$ and any vertex of $X_i \setminus S_{i-1}$ would belong to a triangle with u and v and, hence, would be generated. Two cases must be considered.

  – Let us first assume that $|S_{i-1} \cap X_i| = 1$. In that case, let $u_i \in X_i \setminus S_{i-1}$ maximizing $|N_G[u_i] \setminus V_{i-1}|$ and let $S_i = S_{i-1} \cup \{u_i\}$. Note that S_i generates V_i. We set CONVEX$(i, S_{i-1}) = \{S_i\}$. In order to prove that this choice satisfies all the requirements for the sub-procedure, it is sufficient to prove that there is a generator S^* of G, containing S_{i-1} and with minimum size for this property, such that S^* contains S_i, i.e. such that $u_i \in S^*$.

    Indeed, let S be any generator for G, containing S_{i-1} and with minimum size for this property. By Lemma 22, $S \cap Y_i$ generates V_i, and so, since S_{i-1} does not generate V_i, there must be a vertex $v_i \in Y_i \setminus S_{i-1}$ that is in S. Furthermore, S_i generates V_i, and so, Y_{i-1}. Therefore, the existence of the desired generator S^* follows directly from Lemma 24, by taking $T = Y_i \backslash S_{i-1}$ and $u_i^1 = u_i$.

  – Second, let us assume that $S_{i-1} \cap X_i = \emptyset$. Let $u_i \in X_i$ be a vertex maximizing $|N_G[u_i] \backslash V_{i-1}|$ and let $u_i' \in X_i \setminus \{u_i\}$ be a vertex maximizing $|N_G[u_i'] \setminus V_{i-1}|$ (recall that $X_i \cap S_{i-1} = \emptyset$, so, $u_i, u_i' \notin S_{i-1}$). Ideally, we would like to set CONVEX$(i, S_{i-1}) = \{S_i\}$ in this case, where this time $S_i = S_{i-1} \cup \{u_i, u_i'\}$. However, we were not able to prove that it is always possible, and so, there are several subcases to be considered.

    More precisely, suppose that there exists a generator S for G that is of minimum-size among all those containing S_{i-1} and that contains at least two vertices of $Y_i \backslash S_{i-1}$. In this situation, it can be found by applying Lemma 24 to S, $T = Y_i \backslash S_{i-1}$ and $u_i^1 = u_i, u_i^2 = u_i'$ a generator S^* of the same size as S such that $S_{i-1} \subseteq S^*$ and $u_i, u_i' \in S^*$. Choosing CONVEX$(i, S_{i-1}) = \{S_i\}$ in this case indeed satisfies all the requirements of the sub-procedure. However, if the existence of such generator S cannot be guaranteed, then another set S'_i than S_i will also need to be output.

* First, let us assume that there are at least two vertices $v_i, v'_i \in X_i \setminus X_{i+1}$ that are not generated by S_{i-1}. However, assume that at least one of v_i and v'_i has no neighbor in S_{i-1}. We claim that in this subcase, there always exists a generator S as defined above. The latter will imply that we can choose CONVEX$(i, S_{i-1}) = \{S_i\}$ in this subcase.
  Indeed, w.l.o.g., say $N_G(v_i) \cap S_{i-1} = \emptyset$. Let S be any generator for G, containing S_{i-1} and with minimum size for these properties. By Lemma 21, and since $N_G(v_i) \cap S_{i-1} = \emptyset$, either $v_i \in S \setminus S_{i-1}$ or it has two adjacent neighbors in $S \setminus S_{i-1}$. Furthermore, there must be $w'_i \in N_G[v'_i] \setminus S_{i-1}$ such that $w'_i \in S$ (in order to generate v'_i), and in particular, since $N_G(v_i) \cap S_{i-1} = \emptyset$ and so, no two adjacent vertices in $S_{i-1} \cup \{v_i\}$ are common neighbors of vertex v'_i, we may assume by the above paragraph and Lemma 21 that $w'_i \neq v_i$. Consequently, there must be two distinct vertices $w_i \in N_G[v_i] \setminus S_{i-1}$ and $w'_i \in N_G[v'_i] \setminus S_{i-1}$ such that $w_i, w'_i \in S$ that proves the claim.
* Second, assume that every vertex of $X_i \setminus X_{i+1}$ that is not generated by S_{i-1} has a neighbor in S_{i-1}. Note that since G is an interval graph, there is a vertex of S_{i-1} that is a common neighbor to all vertices in X_i with a neighbor in S_{i-1}. Amongst all vertices in X_i with one neighbor in S_{i-1}, choose one vertex u''_i maximizing $|N_G[u''_i] \setminus V_{i-1}|$ (possibly, $u_i = u''_i$ or $u'_i = u''_i$). In this subcase, we will consider the two possibilities $S'_i = S_{i-1} \cup \{u''_i\}$ or $S_i = S_{i-1} \cup \{u_i, u'_i\}$ and we set CONVEX$(i, S_{i-1}) = \{S_i, S'_i\}$. Note that any of the two choices S_i or S'_i will generate the set V_i. Indeed, this is clear for the case $S_i = S_{i-1} \cup \{u_i, u'_i\}$. In the case $S'_i = S_{i-1} \cup \{u''_i\}$, it is sufficient to notice that every vertex of $X_i \setminus X_{i+1}$ can be generated with u''_i and any vertex of S_{i-1} which is a common neighbor of all vertices of $X_i \setminus X_{i+1}$. Furthermore, in order to prove that our choice for CONVEX$(i, S_{i-1}) = \{S_i, S'_i\}$ satisfies all the requirements for the sub-procedure, it is sufficient to prove that there is a generator S^* of G, containing S_{i-1} and with minimum size for this property, such that $u_i, u'_i \in S^*$ or $u''_i \in S^*$.
  Let S be any generator for G, containing S_{i-1} and with minimum size for this property. As said before, if S contains at least two vertices of $Y_i \setminus S_{i-1}$, then we are done, as in this situation there exists a generator S^* of G, containing S_{i-1} and with minimum size for this property, such that $u_i, u'_i \in S^*$. So, let us assume that S contains no more than one vertex of $Y_i \setminus S_{i-1}$. Since by Lemma 22, $S \cap Y_i$ generates V_i and v_i is not generated by S_{i-1}, there is exactly one vertex $v_i \in Y_i \setminus S_{i-1}$ contained in S. We claim that v_i has a neighbor in S_{i-1}. The latter will imply by Lemma 24 (applied to S, $T = Y_i \cap N_G(S_{i-1})$, $u^1_i = u''_i$) the existence of a generator S^* containing u''_i. In order to prove the claim, let $w_i \in X_i \setminus X_{i+1}$ be a vertex not generated by S_{i-1}. By the hypothesis, w_i has a neighbor in S_{i-1}. In particular, if $w_i \in S$, then we are done as in this situation, $w_i = u''_i$ has a neighbor in S_{i-1}. Otherwise, by Lemmas 21 and 22 w_i has two adjacent neighbors $y, z \in S \cap Y_i$, at most one of which is in S_{i-1} (else, w_i would be generated by S_{i-1}). Since v_i is the unique vertex of $Y_i \setminus S_{i-1}$ that is in S, it implies that v_i is one of y and z, say w.l.o.g. $y = v_i$, and $z \in S_{i-1}$ is a neighbor of v_i, that proves the claim.
* Else, in all other cases, there is a unique vertex $v_i \in X_i \setminus X_{i+1}$ that cannot be generated with S_{i-1}, and this vertex has no neighbor in S_{i-1}. In this subcase, we will consider

the two possibilities $S'_i = S_{i-1} \cup \{v_i\}$ and $S_i = S_{i-1} \cup \{u_i, u'_i\}$ and we will choose CONVEX$(i, S_{i-1}) = \{S_i, S'_i\}$. Again, note that both choices of S_i or S'_i will generate V_i. Furthermore, and as in the previous sub-cases, in order to prove that our choice for CONVEX$(i, S_{i-1}) = \{S_i, S'_i\}$ satisfies all the requirements for the sub-procedure, it is sufficient to prove that there is a generator S^* of G, containing S_{i-1} and with minimum size for these properties, such that $u_i, u'_i \in S^*$ or $v_i \in S^*$. Indeed, let S be any generator for G, containing S_{i-1} and with minimum size for this property. By the Lemma 21, $v_i \in S$ (in such case, it is trivially generated), or it has two adjacent neighbors in S. Moreover, in the latter case, these two neighbors are in $Y_i \setminus S_{i-1}$, which as explained before implies by Lemma 23 the existence of a generator S^* such that $u_i, u'_i \in S^*$.

From the properties of sub-procedure CONVEX, the dynamic programming algorithm presented at the beginning of the proof actually computes a minimum-size generator of G. $\square$

Having explained the guiding principles of the algorithm with Lemma 25, we finally explain how to turn it into a cubic-time algorithm.

Proof Proof of Theorem 20: Computing CONVEX(i, S_{i-1}) for all possible pairs i, S_{i-1} would result in a complexity that is exponential in n. In order to obtain the desired polynomial-time complexity, we will partition the family of sets S_{i-1} into classes such that, when S_{i-1} and S'_{i-1} are in the same class, the set of vertices added by the algorithm is the same. Formally, S_{i-1} and S'_{i-1} will be said equivalent whenever $\{T_i \subseteq Y_i \setminus S_{i-1} \mid S_{i-1} \cup T_i \in \text{CONVEX}(i, S_{i-1})\} = \{T'_i \subseteq Y_i \setminus S'_{i-1} \mid S'_{i-1} \cup T'_i \in \text{CONVEX}(i, S'_{i-1})\}$. By doing so, we will show that we only need to consider polynomially many classes. Let us define $\alpha(S_{i-1}) = \langle j_v, j_e \rangle$, where j_v, respectively j_e, is the maximum index j such that X_j contains a vertex, respectively two adjacent vertices (an edge), of S_{i-1}. We set $j_v = 0$ if $S_{i-1} = \emptyset$, similarly $j_e = 0$ if $G[S_{i-1}]$ is edgeless.

In what follows, we shall prove that all decisions made by the algorithm CONVEX only depend on $\alpha(S_{i-1})$. In other words, for every pair S_{i-1}, S'_{i-1}, generators of V_{i-1}, such that $\alpha(S_{i-1}) = \alpha(S'_{i-1})$ we have $\{T_i \subseteq Y_i \setminus S_{i-1} \mid S_{i-1} \cup T_i \in \text{CONVEX}(i, S_{i-1})\} = \{T'_i \subseteq Y_i \setminus S'_{i-1} \mid S'_{i-1} \cup T'_i \in \text{CONVEX}(i, S'_{i-1})\} = \texttt{generator}(i, \alpha(S_{i-1}))$, where `generator` is a function defined in what follows.

By doing so, we will be able to modify the exponential-time algorithm of Lemma 25 as follows: iteratively, for $i = 1$ to l, the new algorithm computes a set $\mathcal{S}_i$ of subsets of vertices such that: as before for any $S \in \mathcal{S}_i$, $S \subseteq Y_i$ and S generates V_i, *and in addition* $\alpha(S) \neq \alpha(S')$ for every $S, S' \in \mathcal{S}_i$. In more details, at first all possible subsets S of $Y_1 = X_1$ that generates V_1 are considered, and their respective class $\alpha(S)$ is stored. For every such a class $\langle j_v, j_e \rangle$, exactly *one* minimum-size subset S is placed in $\mathcal{S}_1$ such that $\alpha(S) = \langle j_v, j_e \rangle$ and V_1 is generated by S. Then, assuming that $\mathcal{S}_i$ has been built, the next set $\mathcal{S}_{i+1}$ is computed as follows. For any set $S \in \mathcal{S}_i$, the sub-procedure CONVEX(i, S) is applied and it returns one or two subsets, stored in an intermediate set $\mathcal{S}'_i$. Finally, for every class $\langle j_v, j_e \rangle$ of a subset in $\mathcal{S}'_i$, exactly *one* minimum-size subset $S \in \mathcal{S}'_i$ is placed in $\mathcal{S}_i$ such that $\alpha(S) = \langle j_v, j_e \rangle$. Finally, the algorithm outputs the minimum size of a set in $\mathcal{S}_l$ and a corresponding set.

The remaining of the proof is devoted to the description of the sub-procedure `generator`. As before, we prove its correctness simultaneously, *i.e.*, for any S_{i-1} generator of V_{i-1} we have CONVEX$(i, S_{i-1}) = \{T_i \cup S_{i-1} \mid T_i \in \texttt{generator}(i, \alpha(S_{i-1}))\}$.

Let $\alpha(S_{i-1}) = \langle j_v, j_e \rangle$ be defined as above. Note that since G is an interval graph, any vertex of X_i with some neighbor in S_{i-1} must have a neighbor in $S_{i-1} \cap X_{j_v}$, and similarly every vertex of X_i with two adjacent neighbors in S_{i-1} must have two neighbors in $S_{i-1} \cap X_{j_e}$. In what follows, we will rely on the two above observations.

For every vertex $v \in V(G)$, let js_v, jl_v be respectively the smallest and the largest index j such that $v \in X_j$. We will show that we can base the decisions of the algorithm on the js_v (*i.e.*, which case or subcase applies), while we can pick the desired vertices u_i, u_i', u_i'' (as defined earlier in the proof of Lemma 25) by using the jl_v. W.l.o.g. (up to a preprocessing step in $\mathcal{O}(n+m)$-time), the vertices of X_i, resp. the vertices of $X_i \setminus X_{i+1}$, are ordered by increasing value of js_v. Furthermore, for every $1 \leq i \leq l$, we store a vertex f_i such that $js_{f_i} \leq i$ and jl_{f_i} is maximized. Similarly, we store a vertex $f_i' \neq f_i$ such that $js_{f_i'} \leq i$ and $jl_{f_i'}$ is maximized. In this situation, the vertices u_i, u_i', u_i'' (as defined earlier in the proof of Lemma 25) can be chosen respectively as f_i, f_i', f_{j_v}, and accessed in constant-time.

Now let us revisit the three main cases of the sub-procedure CONVEX, as follows.

- First, we need to verify whether S_{i-1} generates V_i. If $j_e \geq i$, then there are two adjacent vertices of S_{i-1} in X_i, and so, V_i is generated by S_{i-1}. Otherwise, since S_{i-1} is assumed to generate V_{i-1}, we have that V_i is generated by S_{i-1} if and only if for every $v_i \in X_i \setminus X_{i+1},\ j_e \in [js_{v_i}, jl_{v_i}]$. Altogether, V_i is generated by S_{i-1} if and only if for all $v_i \in X_i \setminus X_{i+1},\ js_{v_i} \leq j_e$. It can be verified in constant-time by taking $v_i \in X_i \setminus X_{i+1}$ maximizing js_{v_i}.

  We recall that in this first case of the sub-procedure, CONVEX$(i, S_{i-1}) = \{S_{i-1}\}$. Therefore, we set $\texttt{generator}(i, \alpha(S_{i-1})) = \{\emptyset\}$.

- Else, V_i is not generated by S_{i-1}, and so, $|X_i \cap S_{i-1}| \leq 1$. We need to decide whether $X_i \cap S_{i-1} \neq \emptyset$. In this situation, $|X_i \cap S_{i-1}| = 1$ if and only if $i \leq j_v$.

  In this case, we have CONVEX$(i, S_{i-1}) = \{S_i\}$ with $S_i = S_{i-1} \cup \{u_i\}$. In particular, we can set $\texttt{generator}(i, \alpha(S_{i-1})) = \{\{u_i\}\}$. By the choice of u_i (maximizing $|N_G(u_i) \setminus V_{i-1}|$), $jl_{u_i} \geq j_v$. Furthermore, u_i has a neighbor in S_{i-1}, and so, in $S_{i-1} \cap X_{j_v}$. Hence, $\alpha(S_i) = \langle jl_{u_i}, j_v \rangle$ and the class of the new subset S_i can be computed in constant-time.

- Else, $X_i \cap S_{i-1} = \emptyset$ and we need to distinguish between three subcases. As stated above, the vertices $v_i \in X_i \setminus X_{i+1}$ that are not generated by V_{i-1} are precisely those satisfying $j_e \leq js_{v_i}$. By considering the two vertices of $X_i \setminus X_{i+1}$ maximizing js_{v_i}, we can check in constant-time whether there are at least two vertices not generated by S_{i-1}. Finally, for every vertex $v_i \in X_i \setminus X_{i+1}$, it has no neighbor in S_{i-1} if and only if $js_{v_i} \geq j_v$. Again, this can be tested by taking $v_i \in X_i \setminus X_{i+1}$ maximizing js_{v_i}.

  In the first subcase, since CONVEX$(i, S_{i-1}) = \{S_i\}$ with $S_i = S_{i-1} \cup \{u_i, u_i'\}$, we have $\alpha(S_i) = \langle \max\{j_v, jl_{u_i}\}, \max\{j_e, jl_{u_i'}\} \rangle = \langle jl_{u_i}, jl_{u_i'} \rangle$. In particular, we set $\texttt{generator}(i, \alpha(S_i)) = \{\{u_i, u_i'\}\}$.

  In the second subcase, we test for $S_i = S_{i-1} \cup \{u_i, u_i'\}$ and $S_i' = S_{i-1} \cup \{u_i''\}$ and we have CONVEX$(i, S_{i-1}) = \{S_i, S_i'\}$. In particular, we set $\texttt{generator}(i, \alpha(S_i)) = \{\{u_i, u_i'\}, \{u_i''\}\}$. Furthermore, we have $\alpha(S_i') = \langle jl_{u_i''}, j_v \rangle$ since u_i'' has a neighbor in S_{i-1}, and so, in $S_{i-1} \cap X_{j_v}$.

  In the third and final subcase, we test for $S_i = S_{i-1} \cup \{u_i, u_i'\}$ and $S_i' = S_{i-1} \cup \{v_i\}$ and we have CONVEX$(i, S_{i-1}) = \{S_i, S_i'\}$, where $v_i \in X_i \setminus X_{i+1}$ is the only vertex that is not generated by S_{i-1}. In particular, we set $\texttt{generator}(i, \alpha(S_i)) = \{\{u_i, u_i'\}, \{v_i\}\}$. Furthermore, we have $\alpha(S_i') = \langle jl_{v_i}, j_e \rangle$ since v_i has no neighbor in S_{i-1}.

Overall, there are $\mathcal{O}(n^2)$ possibilities for $\alpha(S_{i-1})$. In other words, the main algorithm only need to consider at most $O(n^2)$ subsets in each of the sets $\mathcal{S}_i$. As a result, $in_{cc}(G)$ and a minimum-size generator for G can be computed using dynamic programming in $\mathcal{O}(n^3)$ time and space. □

6 FPT algorithms

This section is devoted to FPT algorithms for computing in_{cc}.

6.1 Bounded treewidth graphs

This section is devoted to the proof that computing the interval number of a graph admits an FPT algorithm parameterized by the treewidth.

Let $G = (V, E)$ be a graph. A *tree-decomposition* $(T, \mathcal{X})$ of G consists of a tree T and a set $\mathcal{X} = (X_t)_{t \in V(T)}$ of subsets of V indexed by the vertices of T, satisfying the following properties.

- $\bigcup_{t \in V(T)} X_t = V$;
- for any edge $\{u, v\} \in E$, there exists $t \in V(T)$ with $u, v \in X_t$;
- for any vertex $v \in V$, the set $\{t \in V(T) \mid v \in X_t\}$ induces a subtree of T.

The *width* of $(T, \mathcal{X})$ equals $\max_{t \in V(T)} |X_t| - 1$ and the *treewidth* of G, denoted by $tw(G)$, is the minimum width over all tree-decompositions of G.

A tree-decomposition $(T, \mathcal{X})$ is *nice* if T is rooted in some vertex $r \in V(T)$, any vertex of T has a at most two children and, for any $t \in V(T)$, one of the following cases holds:

- t is a leaf of T and $|X_t| = 1$ (*Leaf vertex*);
- t has one child u and there exists $v \in V$ such that $X_u = X_t \cup \{v\}$ (*Forget vertex*);
- t has one child u and there exists $v \in V$ such that $X_t = X_u \cup \{v\}$ (*Introduce vertex*);
- t has two children u and w and $X_u = X_w = X_t$ (*Join vertex*).

Theorem 26. *If an n-vertex graph G has tree-width at most k, then $in_{cc}(G)$ can be computed in time $2^{\mathcal{O}(k \log k)} n$.*

Proof: Let $(T, \mathcal{X})$ be a nice tree-decomposition of an n-vertex graph G with width $k = \mathcal{O}(tw(G))$ and $\mathcal{O}(n)$ vertices. Such a decomposition exists and can be computed in time $2^{\mathcal{O}(k)} n$ Bodlaender et al. (2016).

Our algorithm proceeds by dynamic programming from the leaves of T to its root r. For any $t \in V(T)$, let T_t be the subtree of T rooted in t (i.e., if $t = r$ is the root, then $T_t = T$, otherwise T_t is the component of $T \setminus e$ containing t, where e is the edge between t and its parent). Let G_t be the subgraph induced by $\bigcup_{u \in V(T_t)} X_u$.

Representation of a generator S in a bag X_t. First, let us consider a generator $S \subseteq V(G)$ and $t \in V(T)$. We aim at describing how S is "represented" in X_t. Let $n_t = |S \cap V(G_t)|$.

Let $S_1, S_2, \cdots, S_\ell$ be the connected components of S. Because S is a generator and by the properties of tree-decompositions, for any vertex $v \in V(G_t) \setminus (X_t \cup S)$, there exists $i \leq \ell$ such that v has two neighbors in $S_i \cap V(G_t)$.

For any $i \leq \ell$, let $S_{i,1}, \cdots, S_{i,c_i}$ be the connected components induced by $S_i \cap V(G_t)$ in G_t. Then, for any $i \leq \ell$ and $j \leq c_i$, let $S^t_{i,j} = S_{i,j} \cap X_t$. Note that $S^t_{1,1}, \cdots, S^t_{1,c_1}, S^t_{2,1}, \cdots, S^t_{\ell,c_\ell}$ are disjoint subsets of X_t and there are no edges between any two of them.

Let $S^t = \bigcup_{1 \leq i \leq \ell} S_i \cap X_t = S \cap X_t$ and let $N^t = X_t \setminus S^t$ be the set of vertices in X_t that do not belong to the generator S. The set N^t can be partitioned in two subsets. Let A^t be the vertices v of N^t such that, there exists $i \leq \ell$ such that v has two neighbors in $S_i \cap V(G_t)$ (i.e., they are the vertices that are "already" generated). Finally, let $B^t = N^t \setminus A^t$ be the set of vertices that "still need to be generated".

The tuple $(n_t, A^t, B^t, (S^t_{1,1}, \cdots, S^t_{1,c_1}), \cdots, (S^t_{\ell,1}, \cdots, S^t_{\ell,c_\ell}))$ fully characterises the generator S in X_t. This tuple is called the *representative* of S in bag X_t.

Partial solutions. Let $t \in V(T)$. From the previous paragraph, we define a *partial solution at bag X_t* as any tuple $(n_t, A^t, B^t, (S^t_{1,1}, \cdots, S^t_{1,c_1}), \cdots, (S^t_{\ell,1}, \cdots, S^t_{\ell,c_\ell}))$ such that:

- $(A^t, B^t, S^t_{1,1}, \cdots, S^t_{1,c_1}, \cdots, S^t_{\ell,1}, \cdots, S^t_{\ell,c_\ell})$ is a partition of X_t;
- for any four integers a, b, c, d, $(a,b) \neq (c,d)$, no edge has an end in $S^t_{a,b}$ and its other end in $S^t_{c,d}$;
- for any vertex $v \in B^t$, there exists no $i \leq \ell$ such that v has two neighbors in $\bigcup_{j \leq c_i} S^t_{i,j}$.
- $n_t \geq |S^t|$.

The algorithm. Note that, for any $t \in V(T)$, there are at most $2^{\mathcal{O}(k \log k)}$ such partial solutions to be considered (indeed, for each possible partition, we only keep the corresponding tuple for which n_t is minimum).

For any leaf t of T, computing the partial solutions in t (each one is a partition of X_t) can be done in constant time (by brute force), imposing that, in a leaf t, $n_t = |S^t|$.

It remains to prove that, in constant time, we can compute the partial solutions for an internal node t using the partial solutions of its children. There are three cases.

t is a Forget vertex Let $u \in V(T)$ be its child and $v \in V$ such that $X_u = X_t \cup \{v\}$.

Let $P_u = (n_u, A^u, B^u, (S^u_{1,1}, \cdots, S^u_{1,c_1}), \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}))$ be any partial solution computed for bag X_u.

- if $v \in B^u$, then the solution is discarded. Indeed, vertex v will never be generated by a solution extending P_u.
- If there exists $i \leq \ell$ such that $c_i > 1$ and there exists $j \leq c_i$ such that $S^u_{i,j} = \{v\}$, then the solution is discarded. Indeed, the sets $S^u_{i,1}, \cdots, S^u_{i,c_i}$ will never be subsets of a connected component of a generator (there is no way to "reconnect" $S^u_{i,j}$).
- Otherwise, the tuple $(n_u, A^u \setminus \{v\}, B^u, (S^u_{1,1} \setminus \{v\}, \cdots, S^u_{1,c_1} \setminus \{v\}), \cdots, (S^u_{\ell,1} \setminus \{v\}, \cdots, S^u_{\ell,c_\ell} \setminus \{v\}))$ is a partial solution for bag X_t.

Then, for each partial solution with same partition of X_t, we keep the corresponding tuple that has the minimum n_t.

t **is an Introduce vertex** Let $u \in V(T)$ be its child and $v \in V$ such that $X_t = X_u \cup \{v\}$.

Let $P_u = (n_u, A^u, B^u, (S^u_{1,1}, \cdots, S^u_{1,c_1}), \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}))$ be any partial solution computed for bag X_u. There are two ways to include v in a partial solution. That is, any partial solution for bag X_u leads to one or more partial solutions for bag X_t.

First, v may not be part of the generator.

- If there is no $i \leq \ell$ such that v has two neighbors in $\bigcup_{1\leq j\leq c_i} S^u_{i,j}$, then the tuple $(n_u, A^u, B^u \cup \{v\}, (S^u_{1,1}, \cdots, S^u_{1,c_1}), \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}))$ is a partial solution for bag X_t.
- Otherwise, the tuple $(n_u, A^u \cup \{v\}, B^u, (S^u_{1,1}, \cdots, S^u_{1,c_1}), \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}))$ is a partial solution for bag X_t.

Second, it may be possible to include v in the generator.

- Suppose that there is a unique $i \leq \ell$ such that v has neighbors in $\bigcup_{1\leq j\leq c_i} S^u_{i,j}$. In that case, let $I \subseteq \{1, \cdots, c_i\}$ be such that v has neighbors in $S^u_{i,j}$ if and only if $j \in I$. W.l.o.g. (up to reorder the sets), $i = 1$ and $I = \{1, \cdots, h\}$ for some $h \leq c_1$ and let $S^t_{1,1} = \{v\} \cup \bigcup_{j\leq h} S^u_{1,j}$. Moreover, let $X \subseteq B^u$ be the set of neighbors of v that have another neighbor in $\bigcup_{1\leq j\leq c_1} S^u_{1,j}$. Then, the tuple

  $$(n_u + 1, A^u \cup X, B^u \setminus X, (S^t_{1,1}, S^u_{1,h+1}, \cdots, S^u_{1,c_1}), (S^u_{2,1}, \cdots, S^u_{2,c_2}) \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}))$$

  is a partial solution for bag X_t.
- Otherwise, if v has no neighbors in $S^t = \bigcup_{1\leq i\leq \ell} S_i \cap X_t$, then there are several partial solutions that are created:
- – $(n_u + 1, A^u, B^u, (S^u_{1,1}, \cdots, S^u_{1,c_1}), \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}), (\{v\}))$ is a partial solution for bag X_t;
- – for any $i \leq \ell$,

  $$(n_u{+}1, A^u{\cup}X, B^u{\setminus}X, (S^u_{1,1}, \cdots, S^u_{1,c_1}), \cdots, (S^u_{i,1}, \cdots, S^u_{i,c_i}, \{v\}), \cdots, (S^u_{\ell,1}, \cdots, S^u_{\ell,c_\ell}))$$

  is a partial solution for bag X_t, where X is the set of vertices in $B^u \cap N(v)$ that have a neighbor in $\bigcup_{1\leq j\leq c_i} S^u_{i,j}$.

  Then, for each partial solution with same partition of X_t, we keep the corresponding tuple that has the minimum n_t.

t **is a Join vertex** Let $u, w \in V(T)$ be its children.

Let $P_u = (n_u, A^u, B^u, (S^u_{1,1}, \cdots, S^u_{1,c^u_1}), \cdots, (S^u_{\ell_u,1}, \cdots, S^u_{\ell_u,c^u_{\ell_u}}))$ be any partial solution computed for bag X_u and let $P_w = (n_w, A^w, B^w, (S^w_{1,1}, \cdots, S^w_{1,c^w_1}), \cdots, (S^w_{\ell_w,1}, \cdots, S^w_{\ell_w,c^w_{\ell_w}}))$ be any partial solution computed for bag X_w such that

- $\ell_u = \ell_w = \ell$;

- for any $i \leq \ell$, $\bigcup_{j \leq c_i^u} S_{i,j}^u = \bigcup_{j \leq c_i^w} S_{i,j}^w$.

Let $i \leq \ell$. Recall that $(S_{i,1}^u, \cdots, S_{i,c_i^u}^u)$ is supposed to represent some connected component S_i of some expected generator S for G. Precisely, $\bigcup_{j \leq c_i^u} S_{i,j}^u = S \cap X_u$ and, for any $j \leq c_i^u$, $S_{i,j}^u$ is the intersection of a connected component of $S_i \cap V(G_u)$ with $X_u = X_t$. Similarly, for any $j \leq c_i^w$, $S_{i,j}^w$ is the intersection of a connected component of $S_i \cap V(G_w)$ with $X_w = X_t$. To obtain a partial solution in bag X_t, we need to "merge" some components $S_{i,j}^u$ and $S_{i,j'}^w$ if they are part of the same connected component of $S_i \cap V(G_t)$. We proceed as follows.

For any $i \leq \ell$, let us consider the bipartite graph G_i with vertices $\{S_{i,1}^u, \cdots, S_{i,c_i^u}^u, S_{i,1}^w, \cdots, S_{i,c_i^w}^w)$ and, for any $j \leq c_i^u$ and $j' \leq c_{j'}^w$, add an edge between $S_{i,j}^u$ and $S_{i,j'}^w$ if both sets intersect.

Let $C_1, \cdots, C_{c_i^t}$ be the connected components of G_i. For any $j \leq c_i^t$, let $S_{i,j}^t = \bigcup_{S_{i,h}^u \in C_j} S_{i,h}^u \cup \bigcup_{S_{i,h}^w \in C_j} S_{i,h}^w$.

Finally, let $n_t = n_u + n_w - |S^u|$.

Then, $(n_t, A^u \cup A^w, B^u \cap B^w, (S_{1,1}^t, \cdots, S_{1,c_1^t}^t), \cdots, (S_{\ell,1}^t, \cdots, S_{\ell,c_\ell^t}^t))$ is a partial solution for bag X_t.

Then, for each partial solution with same partition of X_t, we keep the corresponding tuple that has the minimum n_t.

Correctness. First, we need to prove that, for any minimum-size generator S for G, the algorithm returns its representative as a partial solution of the root r of the tree-decomposition. For this purpose, we can prove the following more general claim by a classical induction (from the leaves to the root).

Claim 27. *Let S be a minimum-size generator for G. For any $t \in V(T)$, the algorithm computes a representative of S as a partial solution of bag X_t.*

On the other hand, we can prove the following claim by considering the sets of partial solutions (for each bag X_t) that lead to some partial solution computed for the root.

Claim 28. *Any partial solution $(n_r, A^r, B^r, (S_{1,1}^r, \cdots, S_{1,c_1}^r), \cdots, (S_{\ell,1}^r, \cdots, S_{\ell,c_\ell}^r))$ of the root r such that $B^r = \emptyset$ and $c_i = 1$ for all $i \leq \ell$ is the representative to a generator of size n_r for the entire graph.*

The first constraint ensures that all vertices of G are well generated and the latter constraint ensures that the "connected components" of the expected generator have "actually" been connected. □

Since outer-planar graphs have treewidth at most 2 Bodlaender (1998), we get the following corollary:

Corollary 29. *The interval number (in cycle convexity) can be computed in linear-time in the class of outer-planar graphs.*

For purpose of completeness, we present a weaker (but simpler to prove) theorem.

Theorem 30. *Let k, t be fixed integers. $in_{cc}(G) \leq k$ can be decided in time $O(n)$ in the class of graphs G with treewidth at most t.*

Proof: According to Courcelle's Theorem Courcelle and Mosbah (1993), deciding if $in_{cc}(G) \leq k$ (k being a fixed integer) is FPT in $t + k$ (and linear in $|V(G)|$) in the class of graphs of treewidth at most t (t being also fixed) if the problem is expressible in MSOL.

We propose the following logical formula to express the problem.

$\exists v_1, \cdots, v_k \in V, \exists X_1, \cdots, X_k \subseteq V$
// the vertices v_i are the ones of the generator and the sets X_j are the connected components of the generator.

- $\forall i \leq k$, $\exists j \leq k$, such that $v_i \in X_j$ *// all v_i's are in one set X_j*
- $\forall j \leq k$, $\forall v \in X_j$, $\exists i \leq k$ such that $v = v_i$ *// the X_j's contains only vertices of the generator*
- $\forall j \leq k$, $\forall Y \subset X_j$, $\exists v \in Y$ and $\exists u \in X_j \setminus Y$ such that $\{u, v\} \in E$ *// each X_j's is connected*
- $\forall v \in V$, if $v \notin X_j$ for all $j \leq k$, then $\exists j \leq k$, $\exists u \in X_j$, $\exists w \in X_j$, such that $\{u, v\} \in E$ and $\{w, v\} \in E$ and $w \neq u$. *// all vertices can be generated*

□

6.2 $(q, q-4)$-graphs

Let $q \geq 4$. A $(q, q-4)$-graph $G = (V, E)$ is such that for any $S \subseteq V$, $|S| \leq q$, S induces at most $q - 4$ paths on four vertices Babel and Olariu (1998). In this section, $q \geq 4$ being fixed, we present a linear-time algorithm in order to compute $in_{cc}(G)$ and a minimum-size generator for G, for any $(q, q-4)$-graph G. The following result will be used in what follows.

Proposition 31. *If $G = (V, E)$ has a vertex $v \in V(G)$ of degree one, then $in_{cc}(G) = in_{cc}(G - v) + 1$.*

Proof: Since G is simple and v does not have two distinct neighbors, then v must belong to any generator for G. □

The algorithm we present uses a well-known decomposition of $(q, q-4)$-graphs.

Let G_1 and G_2 be two graphs. $G_1 \cup G_2$ denotes the disjoint union of G_1 and G_2. $G_1 \oplus G_2$ denotes the complete join of G_1 and G_2, i.e. the graph obtained from $G_1 \cup G_2$ by adding an edge between any two vertices $v \in V(G_1)$ and $w \in V(G_2)$. A *spider* $G = (S \cup K \cup R, E)$ is a graph with vertex set $V = S \cup K \cup R$ and edge set E such that:

1. (S, K, R) is a partition of V and R may be empty;
2. the subgraph $G[K \cup R]$ induced by K and R is the complete join $K \oplus R$, and K separates S and R, i.e. any path from a vertex in S and a vertex in R contains a vertex in K;
3. S is a stable set, K is a clique, $|S| = |K| \geq 2$, and there exists a bijection $f : S \longrightarrow K$ such that, for all vertices $s \in S$, either $N(s) \cap K = K - \{f(s)\}$ or $N(s) \cap K = \{f(s)\}$. Roughly speaking, the edges between S and K are either a matching or an anti-matching. In the former case or if $|S| = |K| \leq 2$, G is called *thin*, otherwise G is *thick*.

A graph $G = (S \cup K \cup R, E)$ is a *pseudo spider* if it satisfies only the first two properties of a spider. A graph $G = (S \cup K \cup R, E)$ is a *q-pseudo spider* if it is a pseudo spider and, moreover, $|S \cup K| \leq q$. Note that q-pseudo spiders and spiders are pseudo spiders.

Theorem 32 (Babel and Olariu (1998))**.** *Let $q \geq 4$ and let G be a $(q, q-4)$-graph. Then, one of the following holds:*

1. *$G = G_1 \cup G_2$ is the disjoint union of two $(q, q-4)$-graphs G_1 and G_2, or*
2. *$G = G_1 \oplus G_2$ is the join of two $(q, q-4)$-graphs G_1 and G_2, or*
3. *G is a spider $(S \cup K \cup R, E)$ where $G[R]$ is a $(q, q-4)$-graph, or*
4. *G is a q-pseudo spider $(S \cup K \cup R, E)$ where $G[R]$ is a $(q, q-4)$-graph.*

In Babel et al. (2001), the authors have reinforced the above characterization of Theorem 32 as follows. A module in a graph G is any subset $\mathcal{M} \subseteq V(G)$ such that for any $v \in V(G) \setminus \mathcal{M}$, either $\mathcal{M} \subseteq N_G(v)$ or $\mathcal{M} \cap N_G(v) = \emptyset$. A module $\mathcal{M}$ is strong if it does not overlap any other module, *i.e.*, for any module $\mathcal{M}'$ of G, either one of $\mathcal{M}, \mathcal{M}'$ is contained in the other or $\mathcal{M}$ and $\mathcal{M}'$ do not intersect. The quotient graph of G is the graph with vertex-set the inclusion wise maximal strong modules of G that are proper subsets of $V(G)$ such that there is an edge between $\mathcal{M}$ and $\mathcal{M}'$ when there is an edge of $E(G)$ with an end in each of $\mathcal{M}, \mathcal{M}'$. By Babel et al. (2001), if G is a $(q, q-4)$-graph that is a q-pseudo-spider $(S \cup K \cup R, E)$, then either $R = \emptyset$ or the quotient graph of $G[K \cup S]$ is a split graph.

Theorem 33. *For a fixed q, if G is a $(q, q-4)$-graph, then in_{cc} can be computed in $f(q) \cdot (n+m)$-time for some computable function f.*

Proof: Let G be a $(q, q-4)$-graph. Without loss of generality, G is connected (else, we consider the connected components of G separately). Furthermore, if G has a universal vertex u, then by Theorem 14, $in_{cc}(G) = \gamma(G \setminus u) + 1$. Since $G \setminus u$ is a $(q, q-4)$-graph (because this class is closed by induced subgraph) and so, its dominating number $\gamma(G \setminus u)$ can be computed in linear-time Babel et al. (2001), the minimum-size $in_{cc}(G)$ of a generator for G can also be computed in linear-time in this case. Therefore, let us assume for the remaining of the proof that there is no universal vertex of G. In particular, G has $n \geq 2$ vertices, and by Theorem 4, it is easy to check whether $in_{cc}(G) = 2$ in time $\mathcal{O}(n+m)$. Hence, from now on, we may assume that $in_{cc}(G) \geq 3$. We will consider all the cases of Theorem 32 (but the first one since G is connected).

- Suppose that $G = G_1 \oplus G_2$ is the complete join of the two $(q, q-4)$-graphs G_1 and G_2. Note that since G has no universal vertex, the graphs G_1 and G_2 have $n_1 \geq 2$ and $n_2 \geq 2$ vertices, respectively. In such case, since every vertex of G_1, resp. of G_2, is adjacent to every vertex of G_2, resp. of G_1, two vertices on each side are enough in order to generate G, hence $in_{cc}(G) \leq 4$. Therefore, we are left to characterize when $in_{cc}(G) = 3$. We will prove the following claim.

  Claim 34. $in_{cc}(G) = 3$ *if and only if* $\min\{\gamma(G_1), \gamma(G_2)\} = 2$.

  In order to do so, assume the existence of a generator S of size three for G. Clearly, $|S \cap V(G_i)| \geq 2$ for some $i \in \{1, 2\}$. W.l.o.g., let us assume that S has at least two vertices in G_1. Furthermore, since S is a generator for G, every vertex of $V(G) \setminus S$ has at least two neighbors in S.

  We aim at proving that there exists $D \subseteq S \cap V(G_1)$ of size two that dominates G_1. There are three cases to be considered.

- Case $|S \cap V(G_1)| = 2$. Since $|S \cap V(G_2)| = 1$, each vertex of $G_1 \setminus S$ must have at least one neighbor in $S \cap V(G_1)$ to be generated (otherwise it would have a unique neighbor in S). Hence, $S \cap V(G_1)$ dominates G_1.
- Case $|S \cap V(G_1)| = 3$ (i.e., $S \subseteq V(G_1)$) and S is connected. Let $Y \subset S$ be any subset of size two of S. Every vertex of $G_1 \setminus S$ must have at least two neighbors in S (to be generated). Therefore, every vertex of $G_1 \setminus S$ has at least one neighbor in Y. Hence, Y dominates G_1.
- Case $|S \cap V(G_1)| = 3$ and S is not connected. Since S must have at least one connected component with at least 2 vertices, let x and y be two adjacent vertices of S and let $\{z\} = S \setminus \{x, y\}$. Let $w \in V(G_1) \setminus S$. The only way that w is generated is that x and y are neighbors of w. Hence, x dominates $V(G_1) \setminus \{z\}$. Finally, $\{x, z\}$ dominates G_1.

Conversely, suppose that there exists $i \in \{1, 2\}$ such that $\gamma(G_i) = 2$ (observe that $\gamma(G_i) \geq \gamma(G) > 1$ because G has no universal vertex). W.l.o.g., let us assume that $\gamma(G_1) = 2$. Let $S_1 = \{u, v\}$ be a dominating set of G_1, let $x \in V(G_2)$ be arbitrary and let $S = S_1 \cup \{x\}$. By construction, every vertex of G has at least two neighbors in S, furthermore, S induces a connected subgraph of G, hence S is a generator for G by Lemma 2.

So, the claim is proved and $in_{cc}(G) = 3$ if and only if $\min\{\gamma(G_1), \gamma(G_2)\} = 2$, which can be decided in linear-time Babel et al. (2001).

- Now suppose that $G = (S \cup K \cup R, E)$ is a spider. There are two subcases to be considered.
  - If G is a thin spider, then every vertex in S is a 1-vertex, and so, by Proposition 31, $in_{cc}(G) = |S| + in_{cc}(G \setminus S)$. Since $G \setminus S$ is still a $(q, q-4)$-graph, one can apply recursively our algorithm on $G \setminus S$. Precisely, since every vertex $u \in K$ is a universal vertex of $G \setminus S$, by Theorem 14, $in_{cc}(G \setminus S) = \gamma(G \setminus (S \cup \{u\})) + 1$, that can be computed in linear-time Babel et al. (2001).
  - Else, G is a thick spider. Let us prove that $in_{cc}(G) = 3$ (recall that we have assumed that $in_{cc}(G) \geq 3$). Indeed, since G is a thick spider, $|K| = |S| \geq 3$, and we pick any three vertices in K in order to form the subset X that we will prove to be a generator for G. In order to prove it, since X induces a triangle, it is sufficient to prove that every vertex of $G \setminus X$ has at least two adjacent neighbors in X. First, since $X \subseteq K$ and K is a clique, the latter holds for every vertex of $K \setminus X$. Furthermore, since there is a complete join between K and R, it also holds for every vertex of R. Finally, since every vertex of S has $|K| - 1$ neighbors in K, there is at most one vertex in X that is nonadjacent to a given vertex in S, and so, since $|X| \geq 3$, every vertex of S has at least two neighbors in X. Consequently, X is a generator for G.
- Else, $G = (S \cup K \cup R, E)$ is a q-pseudo-spider. Let us assume that $R \neq \emptyset$ (else, since $|K \cup S| \leq q$, the problem can be solved by using exhaustive search). By Babel et al. (2001) the quotient graph $\mathcal{Q}$ of $G[K \cup S]$ is a split graph, with the strong modules in K and the strong modules in S forming respectively a clique and a stable set in $\mathcal{Q}$. In particular, if K is a strong module (i.e., there is a join between S and K), then $G = G[K] \oplus G[S \cup R]$, and so, we go back to a previous case. So, we assume the existence of two strong modules $K_1, K_2 \subseteq K$. Note that there exists a complete join between K_1 and K_2 (because the modules of K induce a clique in Q).

Roughly, we will show that in order to compute a minimum-size generator for G, it suffices to "guess" its part in $S \cup K$. Precisely, we address the following subproblem: given $X_0 \subseteq S \cup K$, does there exist a subset R^* such that $X_0 \cup R^*$ is a generator for G, and if so, what is the minimum-size of such a subset ? We will show the following claim.

Claim 35. *Given $X_0 \subseteq S \cup K$, it can be decided in linear-time whether it exists a set R^* such that $X_0 \cup R^*$ is a generator for G. Moreover, if it exists, such a set R^* with minimum size can be computed in linear-time.*

Overall, since $|S \cup K| \leq q$, there are at most 2^q subsets X_0 to be considered, and so, since q is fixed, $in_{cc}(G)$ can be computed in linear-time in this case.

In order to prove the claim, consider a minimum-size generator X^* for G. Let $X_0 = X^* \setminus R$. Suppose that $X_0 \neq X^*$ (else, we are done).

First we claim that $X_0 \cup \{r\}$ has to generate S for any choice of vertex $r \in R$. Indeed, let $s \in S \setminus X_0$ be arbitrary. By Lemma 2 there is a path in X^* whose both ends are distinct and adjacent to s. In particular, if this path lies onto X_0, then we are done. Else, as K is a SR-separator, the path goes by two vertices in K, that can be linked through vertex r since there is a complete join between K and R. Therefore, s is generated by $X_0 \cup \{r\}$, that proves the claim.

Furthermore, we claim that it can be assumed that $|X^* \cap R| \leq 2$. Indeed, if it is not the case, then let $r_0 \in X^* \cap R$ be arbitrary, and let $u_1 \in K_1, u_2 \in K_2$. We claim that $X = X_0 \cup \{u_1, u_2, r_0\}$ is a generator for G (note that $|X| \leq |X^*|$). Indeed, by the previous claim it generates S. Moreover, since there is a complete join between K and R, and the strong modules in K form a clique of $\mathcal{Q}$, vertices u_1, u_2, r_0 form a triangle and every vertex of R, resp. of K, is adjacent to u_1 and u_2, resp. to r_0 and at least one of u_1 or u_2 (because they are in different modules). Therefore, these three vertices generate $K \cup R$ by Lemma 2, that finally proves the claim.

Thus, we are left to distinguish between the subcases $|X^* \cap R| = 2$ and $|X^* \cap R| = 1$.

- Suppose that $|X^* \cap R| = 2$. We claim that it can be assumed that $X_0 \cap K = \emptyset$. Indeed, if it is not the case, then assume w.l.o.g. that $X_0 \cap K_1 \neq \emptyset$, and let $u_1 \in K_1 \cap X_0$ and let $r_0, r_1 \in X^* \cap R$. Let $u_2 \in K_2$, it can be shown similarly as for the previous claim that $X = X_0 \cup \{u_2, r_0\}$ is a generator for G. Since $|X| \leq |X^*|$ and $|X \cap R| = 1$, the latter proves the claim, and so, from now on let us assume that $X_0 \cap K = \emptyset$. In this subcase, since K is a SR-separator, the two vertices in $X^* \cap R$ generate R, *i.e.*, $in_{cc}(G[R]) \leq 2$. By Theorem 4, it implies that $|R| = 2$ or the two vertices in $X^* \cap R$ are two universal vertices of $G[R]$. Note that any choice of two universal vertices of $G[R]$ will generate $R \cup K$ in this subcase.
- Else, $X^* \cap R = \{r_0\}$. By Lemma 2, every vertex of $R \setminus r_0$ has two neighbors in X^* (in order to be generated). So, $R = \{r_0\}$, or $|X_0 \cap K| \geq 2$, or r_0 is a universal vertex of $G[R]$. Note that, if $|X_0 \cap K| = 2$, then $X_0 \cup \{r\}$ is a generator for G for any choice of $r \in R$, furthermore, for any choice of a universal vertex r_0 of $G[R]$, $X_0 \cup \{r_0\}$ is a generator for G.

To sum up, given X_0 fixed a candidate for the subset R^* can be chosen as follows. Assume that $|R| \geq 3$, or else since q is fixed, the above subproblem can be solved in constant-time by exhaustive search. If X_0 is a generator for G, then $R^* = \emptyset$. Else, if $|X_0 \cap K| \geq 2$, then $R^* = \{r_0\}$ with

$r_0 \in R$ being arbitrary. Else, if $|X_0 \cap K| = 1$, then we search for a universal vertex in R; if no such vertex exists, then we discard X_0, else $R^* = \{r_0\}$ with r_0 being universal to R. Else, $X_0 \cap K = \emptyset$, we search for the existence of two universal vertices in R; if no such two vertices exist, then we discard X_0, else $R^* = \{r_0, r_1\}$ with r_0, r_1 being two universal vertices of R. In all cases, we discard X_0 when $X_0 \cup R^*$ does not generate G. If it does, then by the above characterization R^* is a minimum-size subset of R such that $X_0 \cup R^*$ is a generator for G.

□

Cographs are the graphs without any induced P_4 (path on four vertices). In other words, they are exactly the $(q, q-4)$-graphs for $q = 4$. Therefore, the following corollary directly follows.

Corollary 36. *If G is a cograph, then $in_{cc}(G)$ can be computed in linear-time.*

6.3 Neighborhood Diversity

This section is devoted to the design of an FPT algorithm for computing in_{cc} parameterized by the *neighborhood diversity*. Two distinct vertices u and v are *twins* if $N(u) \setminus v = N(v) \setminus u$. The *neighborhood diversity* of a graph is k, if its vertex set can be partitioned into k sets $P_1, \ldots, P_k$ such that, for every $i \in \{1, \ldots, k\}$, any pair of vertices $u, v \in P_i$ are twins. This parameter was proposed in Lampis (2012), motivated by the fact that a graph of bounded vertex cover also has bounded neighborhood diversity, and therefore the later parameter can be used to obtain more general results.

Many problems have been shown to be FPT when the parameter is the neighborhood diversity Ganian (2012).

Lemma 37. *Let $G = (V, E)$ be a graph, let S be a minimum-size generator for G and let $P \subseteq V$ be a set of pairwise twins. Then, either $|S \cap P| \le 2$ or $P \subseteq S$.*

Furthermore, in the former case, for every $P^* \subseteq P$ of size $|P^*| = |S \cap P|$, the set $S^* = (S \setminus P) \cup P^*$ is also a generator for G.

Proof: Let $P' \subseteq P$ be of size $|P'| = \min\{2, |P \cap S|\}$ and let $S' = (S \setminus P) \cup P'$.

First, we claim that S' is a generator for $V \setminus S$. Indeed, let $x \notin S$. If $x \in S'$, then we are done. Otherwise, recall that every vertex $x \notin S$ is the unique vertex of a cycle C that is not in S. In particular, by taking the smallest such cycle we can assume w.l.o.g. C to be induced. In this situation, C contains no more than two vertices of P since they are pairwise twins. Furthermore, for any $P'' \subseteq P'$ of size $|P''| = |P' \cap V(C)|$, the subset $V(C \setminus P) \cup P''$ induces a cycle C' (again, this is because the vertices of P are pairwise twins). Therefore, since x is the only vertex of C' not in S', x is generated by S', that proves the claim.

Now there are two cases to be distinguished.

- Suppose that S' is a generator for G. By minimality of S and since $|S'| \le |S|$ by construction, we get $|S'| = |S|$ and so, $|S \cap P| = |P'| \le 2$. Moreover since the vertices of P are pairwise twins, for any $P^* \subseteq P$ of size $|P^*| = |P'|$ there is an automorphism of G mapping P' to P^*. Therefore, $S^* = (S \setminus P) \cup P^*$ is also a generator for G.

- Otherwise, S' does not generate G. The latter implies $|S \cap P| \ge 3$ (otherwise, we could go back to the previous case by setting $P' = P \cap S$). Suppose for sake of contradiction that $P \setminus S \ne \emptyset$. Let $v \in P \setminus S$. Since $|S \cap P| \ge 3$, we further assume $v \notin P'$, *i.e.*, $v \notin S \cup S'$. Furthermore, since $V \setminus S$

is generated by S', there exists a cycle C' such that v is the only vertex of C' not in S'. However, for every $v' \in (P \cap S) \setminus S'$, since v and v' are twins it implies that there is a cycle C'' with vertex-set $V(C \setminus v) \cup \{v'\}$, and so, v' is also generated by S'. Overall, $(V \setminus S) \cup S' \cup ((P \cap S) \setminus S') = V$ is generated by S', thereby contradicting the assumption that S' is not a generator for G. As a result, $P \subseteq S$ in this case.

$\square$

Theorem 38. *If the neighborhood diversity of a connected graph G is at most k, then $in_{cc}(G)$ can be computed in $(4^k \cdot \mathcal{O}(n))$-time.*

Proof: Let $P_i,\ 1 \leq i \leq k$ be a partition of V into k pairwise twins, and let S be a minimum-size generator for G. By Lemma 37, for every $1 \leq i \leq k$ there are at most four possibilities for $|P_i \cap S|$: either $|P_i \cap S| \leq 2$ (three possibilities) or $P_i \subseteq S$. Overall, there are 4^k possibilities for the sequence $|S \cap P_1|, |S \cap P_2|, \ldots, |S \cap P_k|$, and since by Lemma 37 the $p_i = |P_i \cap S|$ vertices of $P_i \cap S$ can be chosen arbitrarily, a minimum-size generator S for G can be computed by exhaustive search over 4^k subsets if a partition into $P_i,\ 1 \leq i \leq k$ is given. Such a partition can be computed in linear time Lampis (2012). $\square$

Acknowledgements

We wish to thank the referees for their careful reading of the first version of this manuscript, and their useful comments.

References

B. Anand, M. Changat, S. Klavžar, and I. Peterin. Convex sets in lexicographic products of graphs. *Graphs and Combinatorics*, pages 1–8, feb 2011. ISSN 0911-0119.

J. Araujo, V. Campos, F. Giroire, N. Nisse, L. Sampaio, and R. Soares. On the hull number of some graph classes. Technical Report RR-7567, INRIA, Sept. 2011a. URL `http://hal.inria.fr/inria-00576581/en/`.

J. Araujo, V. Campos, F. Giroire, L. Sampaio, and R. Soares. On the hull number of some graph classes. *Electronic Notes in Discrete Mathematics*, 38(0):49–55, 2011b. ISSN 1571-0653. The Sixth European Conference on Combinatorics, Graph Theory and Applications, EuroComb 2011.

J. Araujo, V. Campos, F. Giroire, N. Nisse, L. Sampaio, and R. Soares. On the hull number of some graph classes. *Theor. Comput. Sci.*, 475:1–12, Mar. 2013. ISSN 0304-3975.

J. Araujo, G. Morel, L. Sampaio, R. Soares, and V. Weber. Hull number: P5-free graphs and reduction rules. *Discrete Applied Mathematics*, 210:171 – 175, 2016. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/j.dam.2015.03.019. URL `http://www.sciencedirect.com/science/article/pii/S0166218X15001687`. LAGOS'13: Seventh Latin-American Algorithms, Graphs, and Optimization Symposium, Playa del Carmen, México — 2013.

J. Araujo, V. Campos, D. Girão, J. Nogueira, A. Salgueiro, and A. Silva. Cycle convexity and the tunnel number of links, 2018. In preparation.

R. T. Araújo, R. M. Sampaio, and J. L. Szwarcfiter. The convexity of induced paths of order three. *Electronic Notes in Discrete Mathematics*, 44:109 – 114, 2013.

L. Babel and S. Olariu. On the isomorphism of graphs with few p4s. In *Proceedings of the 21st International Workshop on Graph-Theoretic Concepts in Computer Science*, WG '95, pages 24–36, London, UK, 1995. Springer-Verlag. ISBN 3-540-60618-1.

L. Babel and S. Olariu. On the structure of graphs with few p4s. *Discrete Appl. Math.*, 84:1–13, May 1998. ISSN 0166-218X.

L. Babel and S. Olariu. On the p-connectedness of graphs - a survey. *Discrete Applied Mathematics*, 95 (1-3):11–33, 1999. ISSN 0166-218X.

L. Babel, T. Kloks, J. Kratochvil, D. Kratsch, H. Müller, and S. Olariu. Efficient algorithms for graphs with few p_4's. *Discrete Mathematics*, 235(1):29–51, 2001.

G. Bacsó and Z. Tuza. Dominating cliques in p_5-free graphs. *Periodica Mathematica Hungarica*, 21(4): 303–308, 1990.

P. Balister, B. Bollobás, J. R. Johnson, and M. Walters. Random majority percolation. *Random Structures & Algorithms*, 36(3):315–340, 2010. ISSN 1098-2418. doi: 10.1002/rsa.20281. URL `http://dx.doi.org/10.1002/rsa.20281`.

J. Balogh and B. Bollobás. Sharp thresholds in bootstrap percolation. *Physica A: Statistical Mechanics and its Applications*, 326(3):305 – 312, 2003. ISSN 0378-4371. doi: http://dx.doi.org/10.1016/S0378-4371(03)00364-9. URL `http://www.sciencedirect.com/science/article/pii/S0378437103003649`.

J. Balogh and G. Pete. Random disease on the square grid. *Random Structures & Algorithms*, 13(3-4):409–422, 1998. ISSN 1098-2418. doi: 10.1002/(SICI)1098-2418(199810/12)13:3/4⟨409::AID-RSA11⟩3.0.CO;2-U. URL `http://dx.doi.org/10.1002/(SICI)1098-2418(199810/12)13:3/4<409::AID-RSA11>3.0.CO;2-U`.

J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. *The IMA Volumes in Mathematics and its Applications*, 56:1–29, 1993.

H. L. Bodlaender. A partial *k*-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209 (1-2):1–45, 1998.

H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.

J. A. Bondy and U. S. R. Murty. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2008.

C. Bujtás and S. Jaskó. Bounds on the 2-domination number. *Discrete Applied Mathematics*, 2017. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/j.dam.2017.05.014. URL `http://www.sciencedirect.com/science/article/pii/S0166218X1730272X`.

J. Cáceres, C. Hernando, M. Mora, I. M. Pelayo, and M. L. Puertas. On the geodetic and the hull numbers in strong product graphs. *Computers and Mathematics with Applications*, 60(11):3020–3031, 2010. ISSN 0898-1221.

G. B. Cagaanan and S. R. Canoy Jr. On the hull sets and hull number of the cartesian product of graphs. *Discrete Mathematics*, 287(1-3):141–144, 2004. ISSN 0012-365X.

V. Campos, R. M. Sampaio, A. Silva, and J. L. Szwarcfiter. Graphs with few p4's under the convexity of paths of order three. *Discrete Applied Mathematics*, 192:28 – 39, 2015. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/j.dam.2014.05.005. URL `http://www.sciencedirect.com/science/article/pii/S0166218X14002078`. 11th Cologne/Twente Workshop on Graphs and Combinatorial Optimization (CTW 2012).

C. C. Centeno, S. Dantas, M. C. Dourado, D. Rautenbach, and J. L. Szwarcfiter. Convex partitions of graphs induced by paths of order three. *Discrete Mathematics & Theoretical Computer Science*, 12(5): 175–184, 2010.

M. Changat and J. Mathew. On triangle path convexity in graphs. *Discrete Mathematics*, 206(1-3):91–95, 1999. ISSN 0012-365X.

M. Changat, H. M. Mulder, and G. Sierksma. Convexities related to path properties on graphs. *Discrete Mathematics*, 290(2-3):117–131, 2005. ISSN 0012-365X.

G. Chartrand, F. Harary, and P. Zhang. On the geodetic number of a graph. *Networks*, 39(1):1–6, 2002a. ISSN 1097-0037.

G. Chartrand, C. E. Wall, and P. Zhang. The convexity number of a graph. *Graphs and Combinatorics*, 18:209–217, 2002b. 10.1007/s003730200014.

G. Chartrand, C. E. Wall, and P. Zhang. The convexity number of a graph. *Graphs and Combinatorics*, 18(2):209–217, 2002c.

G. Chartrand, J. F. Fink, and P. Zhang. The hull number of an oriented graph. *International Journal of Mathematics and Mathematical Sciences*, 2003(36):2265–2275, 2003.

M. Chellali, O. Favaron, A. Hansberg, and L. Volkmann. k-domination and k-independence in graphs: A survey. *Graphs and Combinatorics*, 28(1):1–55, Jan 2012. ISSN 1435-5914. doi: 10.1007/s00373-011-1040-3. URL `https://doi.org/10.1007/s00373-011-1040-3`.

X. Chen and V. Chvátal. Problems related to a de Bruijn-Erdös theorem. *Discrete Appl. Math.*, 156: 2101–2108, June 2008. ISSN 0166-218X.

E. Chiniforooshan and V. Chvátal. A de Bruijn - Erdös theorem and metric spaces. arXiv:0906.0123v1, 2009.

M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264 – 1275, 2008.

B. Clark. The heegaard genus of manifolds obtained by surgery on links and knots. *International Journal of Mathematics and Mathematical Sciences*, 3(3):583–589, 1980.

E. M. M. Coelho, M. C. Dourado, and R. M. Sampaio. *Inapproximability Results for Graph Convexity Parameters*, pages 97–107. Elsevier, 2014.

B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theor. Comput. Sci.*, 109(1&2):49–82, 1993.

I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 624–633. ACM, 2014.

M. C. Dourado, J. G. Gimbel, J. Kratochvíl, F. Protti, and J. L. Szwarcfiter. On the computation of the hull number of a graph. *Discrete Mathematics*, 309(18):5668–5674, 2009. ISSN 0012-365X. Combinatorics 2006, A Meeting in Celebration of Pavol Hell's 60th Birthday (May 1-5, 2006).

M. C. Dourado, F. Protti, D. Rautenbach, and J. L. Szwarcfiter. On the hull number of triangle-free graphs. *SIAM J. Discret. Math.*, 23:2163–2172, January 2010a. ISSN 0895-4801.

M. C. Dourado, F. Protti, and J. L. Szwarcfiter. Complexity results related to monophonic convexity. *Discrete Applied Mathematics*, 158(12):1268–1274, 2010b. Traces from LAGOS'07 IV Latin American Algorithms, Graphs, and Optimization Symposium Puerto Varas - 2007.

M. C. Dourado, D. Rautenbach, V. F. dos Santos, P. M. Schäfer, J. L. Szwarcfiter, and A. Toman. An upper bound on the p_3-radon number. *Discrete Mathematics*, 312(16):2433–2437, 2012.

M. C. Dourado, D. Rautenbach, V. F. dos Santos, P. M. Schäfer, and J. L. Szwarcfiter. On the carathéodory number of interval and graph convexities. *Theor. Comput. Sci.*, 510:127–135, 2013.

P. A. Dreyer and F. S. Roberts. Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, 157(7):1615 – 1627, 2009. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/j.dam.2008.09.012. URL `http://www.sciencedirect.com/science/article/pii/S0166218X08004393`.

P. Duchet. Convex sets in graphs, ii. minimal path convexity. *Journal of Combinatorial Theory, Series B*, 44(3):307 – 316, 1988. ISSN 0095-8956. doi: http://dx.doi.org/10.1016/0095-8956(88)90039-1. URL `http://www.sciencedirect.com/science/article/pii/0095895688900391`.

M. G. Everett and S. B. Seidman. The hull number of a graph. *Discrete Mathematics*, 57(3):217–223, 1985. ISSN 0012-365X.

M. Farber and R. E. Jamison. Convexity in graphs and hypergraphs. *SIAM J. Algebraic Discrete Methods*, 7:433–444, July 1986. ISSN 0196-5212.

A. Farrugia. Orientable convexity, geodetic and hull numbers in graphs. *Discrete Appl. Math.*, 148: 256–262, June 2005. ISSN 0166-218X.

J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1 edition, Mar. 2006. ISBN 3540299521.

F. V. Fomin, P. Golovach, and D. M. Thilikos. Contraction obstructions for treewidth. *Journal of Combinatorial Theory, Series B*, 101(5):302–314, 2011.

R. Ganian. Using neighborhood diversity to solve hard problems. *CoRR*, abs/1201.3091, 2012.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990. ISBN 0716710455.

V. Giakoumakis and J. Vanherpe. On extended P_4-reducible and extended P_4-sparse graphs. *Theoretical Computer Science*, 180:269–286, 1997.

C. Hernando, T. Jiang, M. Mora, I. M. Pelayo, and C. Seara. On the steiner, geodetic and hull numbers of graphs. *Discrete Mathematics*, 293(1–3):139 – 154, 2005a. ISSN 0012-365X. doi: https://doi.org/10.1016/j.disc.2004.08.039. URL `http://www.sciencedirect.com/science/article/pii/S0012365X05000658`. 19th British Combinatorial Conference19th British Combinatorial Conference.

C. Hernando, T. Jiang, M. Mora, I. M. Pelayo, and C. Seara. On the steiner, geodetic and hull numbers of graphs. *Discrete Mathematics*, 293(1-3):139–154, 2005b. ISSN 0012-365X. 19th British Combinatorial Conference.

C. Hoàng. *Perfect graphs*. PhD thesis, McGill University, 1985.

M. Hujter and Z. Tuza. Precoloring extension. ii. graphs classes related to bipartite graphs. *ACTA MATHEMATICA UNIVERSITATIS COMENIANAE*, 62(1):1–11, 1993.

B. Jamison and S. Olariu. A tree representation for p_4-sparse graphs. *Discrete Appl. Math.*, 35(2):115–129, 1992. ISSN 0166-218X.

R. E. Jamison. A perspective on abstract convexity: Classifying alignments by varieties. *Convexity and Related CombinatorialGeometry*, 1982.

Karp, R. M. Reducibility Among Combinatorial Problems. In Miller, R. E. and Thatcher, J. W., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

M. Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64:19–37, 2012. ISSN 0178-4617.

C. Lekkeikerker and J. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51(1):45 – 64, 1962.

J. Liu and H. Zhou. Dominating subgraphs in graphs with some forbidden structures. *Discrete Mathematics*, 135(1–3):163 – 168, 1994. ISSN 0012-365X.

R. M. McConnell and J. P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201:189–241, 1999.

A. Nayak, J. Ren, and N. Santoro. An improved testing scheme for catastrophic fault patterns. *Inf. Process. Lett.*, 73(5-6):199–206, Mar. 2000. ISSN 0020-0190. doi: 10.1016/S0020-0190(00)00012-0. URL `http://dx.doi.org/10.1016/S0020-0190(00)00012-0`.

E. M. Paluga and S. R. Canoy Jr. Monophonic numbers of the join and composition of connected graphs. *Discrete Mathematics*, 307(9):1146 – 1154, 2007. ISSN 0012-365X. doi: http://dx.doi.org/10.1016/j.disc.2006.08.002. URL `http://www.sciencedirect.com/science/article/pii/S0012365X0600625X`.

I. Pelayo. *Geodesic Convexity in Graphs*. SpringerBriefs in Mathematics. Springer New York, 2013. ISBN 9781461486992.

D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282(2):231 – 257, 2002. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/S0304-3975(01)00055-X. URL `http://www.sciencedirect.com/science/article/pii/S030439750100055X`. FUN with Algorithms.

I. Peterin. The pre-hull number and lexicographic product. *Discrete Mathematics*, 312(14):2153–2157, 2012. ISSN 0012-365X. Special Issue: The Sixth Cracow Conference on Graph Theory, Zgorzelisko 2010.

I. F. Ramos, V. F. dos Santos, and J. L. Szwarcfiter. Complexity aspects of the computation of the rank of a graph. *Discrete Mathematics & Theoretical Computer Science*, 16(2):73–86, 2014.

R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970.

M. Tedder, D. G. Corneil, M. Habib, and C. Paul. Simpler Linear-Time Modular Decomposition Via Recursive Factorizing Permutations. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5125 of *Lecture Notes in Computer Science*, pages 634–645, 2008.

J. Varlet. Convexity in tournaments. *Bull. Soc. R. Sci. Liège*, 45:570–586, 1976.

S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.