A New Graph Parameter To Measure Linearity ⋆

Pierre Charbit[a], Michel Habib[a], Lalla Mouatadid[b], Reza Naserasr[a]

[a] *IRIF, CNRS & Université Paris Cité, Paris, France. email: charbit, habib, reza, @irif.fr*
[b] *Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, email: lalla@cs.toronto.edu*

Abstract

Consider a sequence of LexBFS vertex orderings $\sigma_1, \sigma_2, \ldots$ where each ordering σ_i is used to break ties for σ_{i+1}. Since the total number of vertex orderings of a finite graph is finite, this sequence must end in a cycle of vertex orderings. The possible length of this cycle is the main subject of this work. Intuitively, we prove for graphs with a known notion of linearity (e.g., interval graphs with their interval representation on the real line), this cycle cannot be too big, no matter which vertex ordering we start with. More precisely, it was conjectured in [9] that for cocomparability graphs, the size of this cycle is always 2, independent of the starting order. Furthermore [27] asked whether for arbitrary graphs, the size of such a cycle is always bounded by the asteroidal number of the graph. In this work, while we answer this latter question negatively, we provide support for the conjecture on cocomparability graphs by proving it for the subclass of domino-free cocomparability graphs. This subclass contains cographs, proper interval, interval, and cobipartite graphs. We also provide simpler independent proofs for each of these cases which lead to stronger results on this subclasses.

Keywords: Graph search, LexBFS, multisweep algorithms, asteroidal number, cocomparability graphs, interval graphs

1. Introduction

A *graph search* or a *graph traversal* is a mechanism to visit the vertices of a graph. Depth-First Search (DFS) and Breadth-First Search (BFS) are two classical and well studied examples of such traversals. If a graph search visits every vertex exactly once, then it produces a total ordering of the vertices of the graph corresponding to the order in which they are visited. The different searches can be therefore analyzed through the properties of the vertex orderings they produce.

⋆Supported by the ANR-France project HOSIGRA (ANR-17-CE40-0022) and NSERC.

Graph searches are often described by a criterion deciding, given an initial segment of the ordering, which vertex can be placed next. For instance, if we start a BFS at a vertex v, then all the neighbours of v must be visited before the non-neighbours of v. Also, most of the times, there are so called tied vertices, i.e. several vertices that are simultaneously eligible to be placed next, and thus an arbitrary choice can be made. For example in BFS, once the root is chosen, the ordering in which its neighbours are visited can be arbitrary.

Given a graph search, such as BFS, one can thus define a more precise graph search simply by defining tie-breaking rules, and this has proved to be a powerful technique to understand and analyze the structure of certain graph classes. This line of work originally started in 1976 by Rose, Tarjan, and Lueker, when they introduced the lexicographic variant of BFS in [23], known as *lexicographic breadth first search*, or LexBFS for short. One of the first uses of this graph search was the simplest linear time algorithm to recognize chordal graphs [23]. Since then, LexBFS has led to a number of simple, efficient, and elegant algorithms on various graph classes [7, 9, 17].

One way to break *all* ties while constructing an ordering τ consists in using another ordering σ : if there is a tie between two vertices x and y, one shall pick the one that is the "greatest" in σ. This was introduced by Simon in [24] for LexBFS and is known as the $^+$ rule. Given an order σ on the vertices of G, Simon defines LexBFS$^+(G, \sigma)$ as the (unique) LexBFS ordering of the vertices of G obtained by breaking ties by always picking the right most vertex with respect to σ (for instance, LexBFS$^+(G, \sigma)$ starts with the last vertex of σ). Now given an initial ordering σ_0 on the vertices of G, one can thus define a sequence $\sigma_0, \sigma_1, \sigma_2, \ldots$ of orderings on V by setting $\sigma_i = \text{LexBFS}^+(G, \sigma_{i-1})$. This technique is known as a *multisweep algorithm* and has been used to introduce fast recognition algorithms for graph classes such as proper interval, interval, and cocomparability graphs [2, 7, 9]. The idea here is to prove some kind of convergence to say that this process will eventually yield some vertex ordering with strong structural properties. This technique is of course especially relevant for the study of graph classes which are defined, or characterized, by the existence of certain types of vertex orderings. For instance unit interval graphs are defined as intersection graphs of interval of length 1 of the real line, but it is a classical theorem that they are exactly the graphs whose vertex set can be ordered such that for any three vertices a, b, c with $a \prec b \prec c$, $ac \in E$ implies that $ab \in E$ and $bc \in E$. In [2] a very simple certifying recognition algorithm based on LexBFS$^+$ is given : starting from any ordering, 3 sweeps must provide such an order (which is easy to check) if the input graph is unit interval.

Evidently, as the number of distinct vertex orderings of a finite graph is finite, no matter which ordering σ_0 we start with, this sequence $\{\sigma_i\}_{i\geq 1}$ of LexBFS$^+$ orderings will eventually cycle. That is, for some i and k, $\sigma_{i+k} = \sigma_i$. For general graphs this observation raises two interesting questions :

(i) Among all possible choices of σ_0 as a start ordering, how long does it take to reach a cycle?

(ii) How large can this cycle be?

This paper is concerned with these questions for the class of cocomparability graphs, a superclass of interval graphs characterized by the existence of a so called *cocomparability ordering* : for any three vertices a, b, c with $a \prec b \prec c$, $ac \in E$ implies that $ab \in E$ or $bc \in E$ (such an order is a transitive order - i.e. a linear extension of a transitive orientation - of the complement graph, hence the name of the class).

One important reason for restricting our attention to cocomparability graphs is because Dusart and Habib proved the following theorem.

Theorem 1.1. *[9] If G is a cocomparability graph on n vertices, and σ_0 an arbitrary ordering of $V(G)$, define a sequence $\{\sigma_i\}_{i\geq 1}$ of* LexBFS$^+$ *orderings of G as $\sigma_i = \text{LexBFS}(G, \sigma_{i-1})$. Then σ_n is a cocomparability ordering of the vertices of G.*

While this theorem guarantees for cocomparability graphs that a multi-sweep process will reach a cocomparability ordering in at most n iterations, we don't know in general any non-trivial bound on when the cycle will be reached. For some subclasses of cocomparability graphs, we prove such bonds in this paper.

Regarding the second question above (ii), and again restricted to the class of cocomparability graphs, Dusart and Habib [9] have conjectured that, no matter which initial ordering we start with, the length of the cycle is at most 2 (a cycle of length 1 being in fact impossible except for the one vertex graph, since the last vertex of an order is always the first vertex of the next order).

Conjecture 1.2. *Given a cocomparability graph G, an arbitrary ordering σ_0 of $V(G)$, and a sequence $\{\sigma_i\}_{i\geq 1}$ of* LexBFS$^+$ *orderings of G where σ_i is used to break ties for σ_{i+1}, for i sufficiently large, we have $\sigma_i = \sigma_{i+2}$.*

Observing that cocomparability graphs are asteroidal triple-free, and thus have asteroidal number two, Stacho asked if the length of all such cycles is bounded by the asteroidal number of the graph [27] .

In this work, we first answer Stacho's question negatively. Then, we provide strong support for the conjecture of Dusart and Habib by proving it for cocomparability graphs that do not contain a particular 6 vertex graph (called domino) as an induced subgraph. While this subclass of cocomparability graphs contains proper interval graphs, interval graphs, cographs and cobipartite graphs, we additionally give for each of these cases an independent proof which provides stronger results, and sheds light into structural properties of these graph classes.

The structure of the paper is as follows: we finish this introduction section by giving basic definitions and fixing our notations. In Section 2 we give all the necessary background to understand LexBFS properties and its use in multisweep algorithms. We also introduce, define, and discuss LexCycle(G), the main invariant studied in our paper. In particular, we give a construction that gives an answer to the question of Stacho mentioned earlier. In Section 3, we expose various results related to vertex ordering characterizations of the classes of graphs and the graph searches studied in the paper. Section 4 contains our main results mentioned in the previous paragraph about Conjecture 1.2 in the subclass of domino-free cocomparability graphs. Finally in Section 5 we present further ideas, and research directions.

1.1. Notations

A graph G is a pair (V, E) where V is a finite set whose elements are called vertices, and E is a set of unordered pairs of V called edges. We sometimes write $V(G)$ and $E(G)$ to denote the vertices and the edges of a graph G. If no ambiguity occurs, we will always use the letters n and m to denote respectively the number of vertices and edges of a graph G. Given a pair of adjacent vertices u and v, we write uv to denote the edge in E with endpoints u and v. We denote by $N(v) = \{u : uv \in E\}$ the open neighbourhood of vertex v, and $N[v] = N(v) \cup \{v\}$ the closed neighbourhood of v. We write $G[V']$ to denote the *induced subgraph* (V', E') of $G = (V, E)$ on the subset V' of V, where for every pair $u, v \in V', uv \in E'$ if and only if $uv \in E$. A graph class $\mathcal{G}$ is said to be *hereditary* if it is closed under induced subgraphs. The complement of a graph $G = (V, E)$ is the graph $\overline{G}(V, \overline{E})$ where $uv \in \overline{E}$ if and only if $uv \notin E$. A *private neighbour* of a vertex u with respect to a vertex v is a third vertex w that is adjacent to u but not v: $uw \in E, vw \notin E$.

A set $S \subseteq V$ is an independent set if for all $a, b \in S, ab \notin E$, and is a clique set if for all $a, b \in S, ab \in E$. Given a pair of vertices u and v, the distance between u and v, denoted $d(u, v)$, is the length of a shortest u, v path. A *diametral* path of a graph is a shortest u, v path where u and v are at the maximum distance among all pairs of vertices. A *dominating* path in a graph is a path where all the vertices of the graph are either on the path or have a neighbour on the path. A triple of independent vertices u, v, w forms an *asteroidal triple* (AT) if every pair of the triple remains connected when the third vertex and its closed neighbourhood are removed from the graph. In general, a set A of vertices of G forms an *asteroidal set* if for each vertex $a \in A$, the set $A \backslash \{a\}$ is contained in one connected component of $G[V \backslash N[a]]$. The maximum cardinality of an asteroidal set of G, denoted $an(G)$, is called the *asteroidal number* of G. A graph is *AT-free* if it does not contain an asteroidal triple. The class of AT-free graphs contains cocomparability graphs. A *domino* (Fig. 1) is the induced graph $G = (V = \{a, b, c, d, e, f\}, E = \{ab, ac, bd, cd, ce, df, ef\})$.

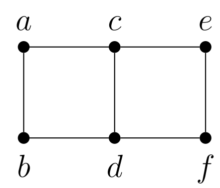


Figure 1: Domino

Let $[k]$ denote the set of integers 1 to k. Given a graph $G = (V, E)$, an *ordering* σ of G is a bijection $\sigma : V \leftrightarrow [n]$. For $v \in V$, $\sigma(v)$ refers to the position of v in σ. For a pair u, v of vertices we write $u \prec_\sigma v$ if and only if $\sigma(u) < \sigma(v)$; we also say that u (resp. v) *is to the left of* (resp. *right of*) v (resp. u). We write $\{\sigma_i\}_{i \geq 1}$ to denote a sequence of orderings $\sigma_1, \sigma_2, \ldots$. We also write $\sigma_{i>1}$ to denote an ordering σ_i where $i > 1$.

Given a sequence of orderings $\{\sigma_i\}_{i \geq 1}$ of a graph G, and an edge $ab \in E$, we write $a \prec_i b$ if $a \prec_{\sigma_i} b$, and $a \prec_{i,j} b$ if $a \prec_i b$ and $a \prec_j b$. Given an ordering $\sigma = v_1, v_2, \ldots, v_n$ of G, we write σ^d to denote the *dual* (also called *reverse*) ordering of σ; that is $\sigma^d = v_n, v_{n-1}, \ldots, v_2, v_1$. For an ordering $\sigma = v_1, v_2, \ldots, v_n$, the interval $\sigma[v_s, \ldots, v_t]$ denotes the ordering of σ restricted to the vertices $\{v_s, v_{s+1}, \ldots, v_t\}$ as numbered by σ. Similarly, if $S \subseteq V$, and σ an ordering of V, we write $\sigma[S]$ to denote the ordering of σ restricted to the vertices of S.

2. LexBFS, multisweep Algorithms and LexCycle

A *multisweep algorithm* is an algorithm that computes a sequence of orderings where each ordering σ_i uses the previous ordering σ_{i-1} to break ties using some predefined tie-breaking rules. We focus on one specific tie-breaking rule: **the $^+$ rule**, formally defined as follows: Given a graph $G = (V, E)$, an ordering σ of G, and a graph search S (such as LexBFS), $S^+(G, \sigma)$ is a new ordering τ of G that uses σ to break any remaining ties from the S search. In particular, given a set T of tied vertices, the $^+$ rule chooses the vertex in T that is rightmost in σ. We sometimes write $\tau = S^+(\sigma)$ instead of $\tau = S^+(G, \sigma)$ if there is no ambiguity on the graph considered.

In this work, we focus on LexBFS based multisweep algorithms. LexBFS is a variant of BFS that assigns lexicographic labels to vertices, and breaks ties between them by choosing vertices with lexicographically highest labels. The labels are words over the alphabet $\{1, ..., n\}$. We denote by label(v) the label of a vertex v. By convention ϵ denotes the empty word. LexBFS was initially introduced by Rose, Tarjan, and Lueker to recognize chordal graphs [23]. We present LexBFS in Algorithm 1 below. The operation *append*$(n - i)$ in Algorithm 1, puts the letter $n - i$ at the end of the word.

Algorithm 1 LexBFS

Input: A graph $G = (V, E)$ and a start vertex s

Output: An ordering σ of V

assign the label ϵ to all vertices, and label$(s) \leftarrow \{n\}$
for $i \leftarrow 1$ to n **do**
    pick an unnumbered vertex v with lexicographically largest label
    $\sigma(v) \leftarrow i$ $\triangleright$ v is assigned the number i
    foreach unnumbered vertex w adjacent to v **do**
        append$(n - i)$ to label(w)
    end for
end for

Starting from an ordering σ_0 of G, a multisweep LexBFS$^+$ process consists of computing the following sequence: $\sigma_{i+1} = \text{LexBFS}^+(G, \sigma_i)$. Since G has a finite number of LexBFS orderings, such a sequence must get into a finite cycle of vertex orderings. This leads to the definition below, notice that there is no assumption on the starting vertex ordering σ_0.

Definition 2.1 (LexCycle)**.** *For a graph $G = (V, E)$, let* LexCycle(G) *be the* ***maximum*** *length of a cycle of vertex orderings obtained via a sequence of LexBFS$^+$ sweeps.*

Note that contrary to other classical invariants, it is not at all clear whether this should be a monotone function for the induced subgraph relation. The following question is still open, even for cocomparability graphs.

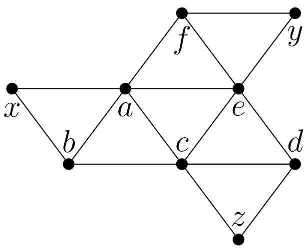
Question 2.2. *If H is an induced subgraph of G, is it true that* LexCycle(H) *is at most* LexCycle(G)*?*

Another viewpoint on LexCycle(G) is obtained by constructing a directed graph G_{lex} whose vertices are all LexBFS orderings of G, and with an arc from σ to τ if $\text{LexBFS}^+(G, \sigma) = \tau$. The digraph G_{lex} is a functional digraph : every vertex has an out-degree of exactly one, and therefore every connected component of G_{lex} is a circuit on which are planted some directed trees. For instance, if K is a clique, K_{lex} is just the union of directed circuits of size two joining one permutation to its reverse. LexCycle(G) is then just the maximum size of a directed circuit in G_{lex}, and we do not know of any example of a graph with two distinct cycle lengths.

In this work, we study the first properties of this new graph invariant, LexCycle. Due to the nature of the $^+$ rule, LexCycle(G) ≥ 2 as soon as G contains more than one vertex (the last vertex of an order is the first vertex of the next one). Obviously LexCycle(G) $\leq n!$, and more precisely LexCycle(G) is bounded by the number of LexBFS orderings of G. We introduce a construction, *Starjoin*, below which suggests (but does not yet prove) that solely based on the number of vertices and without the use of the structural constraints on the graph, we cannot bound LexCycle(G) by a polynomial on n. This construction will allow us, at the end of this section, to answer the question of Stacho [27] mentioned in the introduction, which asks if LexCycle(G) $\leq an(G)$ for any graph.

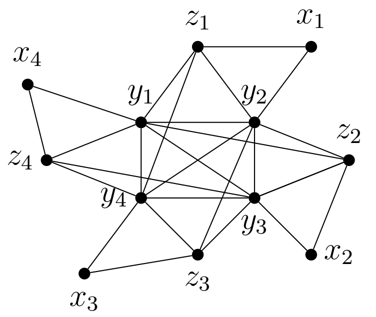
We start by constructing some graphs with LexCycle ≥ 3 :

- G_3 is the graph represented on Figure 2. It satisfies LexCycle(G_3) $\geq 3 = an(G_3)$, as shown by the multisweep starting with $\sigma_1 = x, b, a, c, e, f, d, z, y$.
- G_4 is the graph represented on Figure 3. It satisfies LexCycle(G_4) $\geq 4 = an(G_4)$, as shown by the multisweep starting with $\mu_1 = \text{LexBFS}(G) = x_4, z_4, y_1, y_3, y_4, y_2, z_2, z_1, z_3, x_2, x_3, x_1$.



$$\begin{aligned}
\text{LexBFS}(G) &= \sigma_1 = x, b, a, c, e, f, d, z, y \\
\text{LexBFS}^+(\sigma_1) &= \sigma_2 = y, f, e, a, c, d, b, x, z \\
\text{LexBFS}^+(\sigma_2) &= \sigma_3 = z, d, c, e, a, b, f, y, x \\
\text{LexBFS}^+(\sigma_3) &= \sigma_1 = x, b, a, c, e, f, d, z, y
\end{aligned}$$

Figure 2: Example of a graph with LexCycle(G_3) ≥ 3 where the 3-cycle consists of $C_3 = [\sigma_1, \sigma_2, \sigma_3]$.



$$\begin{aligned}
\mu_1 &= x_4, z_4, y_1, y_3, y_4, y_2, z_2, z_1, z_3, x_2, x_3, x_1 \\
\mu_1^+ \quad = \mu_2 &= x_1, z_1, y_2, y_4, y_1, y_3, z_3, z_2, z_4, x_3, x_4, x_2 \\
\mu_2^+ \quad = \mu_3 &= x_2, z_2, y_3, y_1, y_2, y_4, z_4, z_3, z_1, x_4, x_1, x_3 \\
\mu_3^+ \quad = \mu_4 &= x_3, z_3, y_4, y_2, y_3, y_1, z_1, z_4, z_2, x_1, x_2, x_4 \\
\mu_4^+ \quad = \mu_1 &= x_4, z_4, y_1, y_3, y_4, y_2, z_2, z_1, z_3, x_2, x_3, x_1
\end{aligned}$$

Figure 3: Example of a graph with LexCycle$(G_4) \geq 4$ where the 4-cycle consists of $C_4 = [\mu_1, \mu_2, \mu_3, \mu_4]$.

We now show how one can construct graphs with LexCycle$(G) > an(G)$. Consider the following graph operation that we call *Starjoin*.

Definition 2.3 (Starjoin)**.** *For a family of vertex disjoint connected graphs $\{G_i\}_{1 \leq i \leq k}$, we define $H = Starjoin(G_1, \dots G_k)$ as follows: For $i \in [k]$, add a universal vertex g_i to G_i, then add a root vertex r adjacent to all g_i's.*

Proposition 2.4. *Let G_i be a graph with a cycle C_i in a sequence of* LexBFS$^+$ *orderings of G_i and let $H = Starjoin(G_1, \dots G_k)$. We have*

- $an(H) = \max\{k, an(G_1), an(G_2), \dots, an(G_k)\}$
- LexCycle$(H) \geq \text{lcm}_{1 \leq i \leq k}\{|C_i|\}$*, where* lcm *stands for the least common multiple.*

Proof. Notice first that selecting one vertex per G_i would create a k-asteroidal set. Since every g_i vertex is universal to G_i, we can easily see that every asteroidal set of H is either restricted to one G_i, or it contains at most one vertex per G_i. This yields the first formula.

For the second property, we notice first that a cycle of LexBFS$^+$ orderings is completely determined by its initial LexBFS ordering, since all ties are resolved using the $^+$ rule. For $1 \leq i \leq k$, let σ_1^i denote the first LexBFS$^+$ ordering on C_i, the cycle in a sequence of LexBFS$^+$ orderings of G_i.

Consider the following LexBFS ordering of H: $\sigma_1^H = r, g_1, \dots g_k \sigma_1^1, \dots \sigma_1^k$. Consider the cycle of LexBFS$^+$ orderings that will result after running a sequence of LexBFS$^+$, starting with σ_1^H as its first ordering. Notice that in any LexBFS$^+$ ordering in this cycle, the vertices of G_i are consecutive, with the exception of g_i that can appear in between G_i's vertices. Furthermore $\sigma_j^H[G_i] = \text{LexBFS}^+(G_i, \sigma_{j-1}^i)$. Therefore if we take σ_1^i as the first LexBFS$^+$ ordering of C_i, then the length of the cycle generated by σ_1^H is necessarily a multiple of $|C_i|$. □

We are now ready to answer Stacho's conjecture negatively.

Corollary 2.5. *There exists a graph G satisfying $LexCycle(G) > an(G)$.*

Proof. To see this, consider $H = Starjoin(G_3, G_4)$ constructed using the graphs in Figures 2 and 3. By Proposition 2.4, $an(H) = 4$ and LexCycle$(H) \geq 12$. □

A natural question to raise here is whether LexCycle can be bounded by some function of the asteroidal number. In order to disprove this fact, it would be enough by Proposition 2.4 to generalize the constructions of G_3 and G_4 to graphs with bounded asteroidal number but arbitrarily large prime LexCycle values. We do not have such a generalization yet.

3. Vertex Ordering Characterizations of Classes and Searches

Given a graph class $\mathcal{G}$, a *vertex ordering characterization* (or VOC) of $\mathcal{G}$ is a characterization of a graph class given by the existence of a total ordering on the vertices with specific properties. VOCs have led to a number of efficient algorithms, and are often the basis of various graph recognition algorithms, see for instance [23, 3, 7, 19, 13]. In this section, we describe some of these VOCs for the graph classes for which we will prove the validity of Conjecture 1.2 in the Section 4

A graph $G = (V, E)$ is an *interval graph* if there exists a collection of intervals $(I_v)_{v \in V}$ such that $uv \in E$ if and only if the intervals I_v and I_u have non empty intersection. Given G, such a collection of intervals is not unique and is called an *interval representation* of G. Given an interval representation $\mathcal{R}$, one can canonically obtain two orderings of the vertices of G: a *left endpoint* ordering of $\mathcal{R}$ is an ordering of the intervals by increasing value of their left endpoint, and a *right endpoint* ordering of $\mathcal{R}$ is the ordering of the intervals by decreasing value of their right endpoint. If some intervals have identical left or right endpoint this can be ambiguous, so more precisely a left (resp. right) endpoint ordering of a collection of intervals $([l(v), r(v)])_{v \in V}$ is any ordering $\prec$ of V such that for all $u, v \in V$, $u \prec v$ implies $l(u) \leq l(v)$ (resp $r(u) \geq r(v)$). It is easy to see that any of these orderings satisfy the following VOC that is in fact a characterization of interval graphs: a graph G is an interval graph if and only if there exists an *I-ordering*, that is an ordering σ of G such that :

$$\text{for every triple } a \prec_\sigma b \prec_\sigma c, \text{ if } ac \in E \text{ then } ab \in E$$

It is a characterization of interval graphs since one can indeed prove that any I-ordering is a left endpoint ordering of some interval representation $\mathcal{R}$ of G.

An interval graph is a *proper interval graph* if no interval in the interval representation is fully contained in another interval. Proper interval graphs were shown in [26] to be precisely the interval graphs that admit a representation where all the intervals have unit length, and are therefore also called *unit interval graphs*. They are also characterized by the following VOC : $G = (V, E)$ is a proper interval graph if and only if V admits a *PI-ordering* : an ordering σ such that

for every triple $a \prec_\sigma b \prec_\sigma c$, if $ac \in E$ then $ab \in E$ and $bc \in E$.

This VOC follows from the fact that in proper interval graphs, left endpoint and right endpoint orderings are the same.

A *comparability graph* is a graph $G = (V, E)$ that admits a transitive orientation of its edges. That is, there exists an orientation on $E(G)$, where for any triple of vertices x, y, z, if $xy, yz \in E(G)$ are oriented $x \to y$ and $y \to z$, then the edge xz must exist and is oriented $x \to z$. This transitivity can be captured in a vertex ordering of $V(G)$ known as a *comparability ordering* or a transitive order. In particular, a transitive order is an ordering σ of the vertices of G where if $x \prec_\sigma y \prec_\sigma z$ and $xy, yz \in E$, then $xz \in E$. A *cocomparability graph* is the complement of a comparability graph. This definition thus translates into a VOC : a graph $G = (V, E)$ is a cocomparability graph if V admits a so called *cocomparability ordering* (see [16]), that is an ordering σ of V such that

for any triple $a \prec_\sigma b \prec_\sigma c$, if $ac \in E$ then $ab \in E$ or $bc \in E$

For a graph G with an order σ on its vertices a triple $a \prec_\sigma b \prec_\sigma c$ with $ac \in E$, $ab \notin E$ and $bc \notin E$ is called an *umbrella*, which is why cocomparability orderings are sometimes called *umbrella-free orderings*.

One can easily see from these vertex orderings that:

$$\text{Proper Interval} \subsetneq \text{Interval} \subsetneq \text{Cocomparability}$$

It is moreover proven in [11] that interval graphs are chordal (no induced cycle of length at least 4), and even more : they are exactly the C_4-free cocomparability graphs.

Also, it is proved in [12] that the class of cocomparability graphs are asteroidal triple-free, thus all these graphs have asteroidal number at most two.

Other graph classes we consider in this paper are *domino-free* cocomparability graphs (cocomparability graphs that do not contain the domino as in induced subgraph) and *cobipartite* graphs (the complements of bipartite graphs). Since a domino contain a C_4 and since interval graphs do not, interval graphs are domino-free. Similarly, a domino contains a independent set of size 3, so cobipartite graphs form also a subclass of domino-free comparability graph. All inclusions are represented on Figure 4.

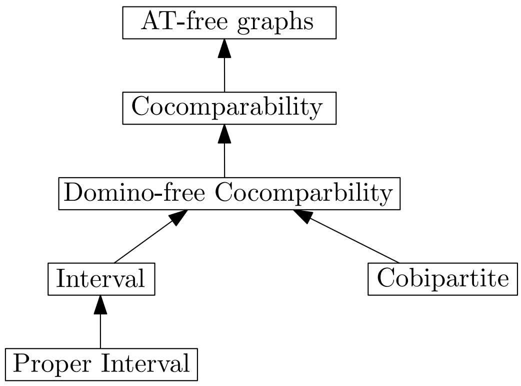


Figure 4: Graph classes studied in this article

Vertex orderings produced by searches can also be characterized by vertex orderings (see [5] for such results). LexBFS in particular has the following VOC, known as the LexBFS *four point condition*.

Theorem 3.1. *[8](LexBFS 4PC) Let* $G = (V, E)$ *be an arbitrary graph. An ordering* σ *is a LexBFS ordering of* G *if and only if for every triple* $a \prec_\sigma b \prec_\sigma c$*, if* $ac \in E, ab \notin E$*, then there exists a vertex* d *such that* $d \prec_\sigma a$ *and* $db \in E, dc \notin E$*.*

We call the triple a, b, c as described in Theorem 3.1 above a *bad triple*. Observe that the vertex d here is private neighbour of b with respect to c. When choosing vertex d as described above, we often choose it as the *left most private neighbour* of b with respect to c in σ and write $d = \text{LMPN}(b|_\sigma c)$. This is to say that prior to visiting vertex d in σ, vertices b and c were tied: every vertex before d in σ is either a common neighbour or a common non-neighbour of b and c (or equivalently label(b) = label(c) as assigned by Algorithm 1), and vertex d caused $b \prec_\sigma c$.

Combining VOCs for graph classes with the LexBFS 4PC has already led to a number of structural results [3, 18, 4]. Here we focus on LexBFS properties on cocomparability graphs. In this case, the 4PC can be refined with a stronger statement that we call C_4 property.

Property 3.2 (The LexBFS C_4 Property)**.** *Let $G = (V, E)$ be a cocomparability graph and σ a LexBFS cocomparability order of V. If σ has a bad LexBFS triple $a \prec_\sigma b \prec_\sigma c$, then there exists a vertex d such that $d \prec_\sigma a$ and G has an induced $C_4 = d, a, b, c$ where $da, db, ac, bc \in E$.*

Proof. To see this, it suffices to use the LexBFS 4PC and the cocomparability VOC properties. Since σ is a cocomparability ordering, and $ab \notin E$ then $bc \in E$. Then, using the LexBFS 4PC, there must exist a vertex $d \prec a$ such that $db \in E, dc \notin E$. Once again since $d \prec a \prec b$ and $db \in E, ab \notin E$, it follows that $da \in E$ otherwise we contradict σ being a cocomparability ordering. □

We add here another lemma with a flavour similar to the 4PC property, that we will use very often when studying LexBFS$^+$ multisweep sequences. Note that it is true for any graph.

Lemma 3.3. *Let G be a graph with an ordering σ of its vertices and let $\tau = \text{LexBFS}^+(G, \sigma)$. If a and b are vertices such that $a \prec_\sigma b$ and $a \prec_\tau b$, then there exists a vertex c with $c \prec_\tau a$ such that $ca \in E$ and $cb \notin E$. Furthermore, if c is the leftmost vertex for this property (i.e. $c = LMPN(a|_\tau b)$), then every vertex that precedes c in τ is either adjacent to both a and b or to none of them.*

Proof. This is just the consequence of the $^+$ rule: if a precedes b in both orderings, then it means a and b were not tied when a was picked during the construction of τ, and therefore the label of a was strictly larger than the one of b, which exactly translates into the conclusion of the Lemma. □

A consequence of the previous lemma is a result from [4] known as the *Flipping Lemma*, that gives an intuition as to why Conjecture 1.2 could be true.

Lemma 3.4 (The Flipping Lemma,[4])**.** *Let $G = (V, E)$ be a cocomparability graph, σ a cocomparability ordering of G and $\tau = LexBFS^+(\sigma)$. For every pair u, v such that $uv \notin E$, $u \prec_\sigma v$ if and only if $v \prec_\tau u$.*

Proof. Assume by contradiction that there exists vertices u and v such that $u \prec_\sigma v$ and $u \prec_\tau v$, and choose such a pair with the left most possible element u with respect to τ. By Lemma 3.3, there exists a vertex w such that $w \prec_\tau u$, $wu \in E$ and $wv \notin E$. Because of the choice of the pair (u, v), we must have $v \prec_\sigma w$, but now the triple (u, v, w) forms an umbrella in σ, which contradicts the fact that σ is a cocomparability order on G. □

Given that a comparability ordering is an umbrella-free ordering, the Flipping Lemma directly implies the following result of [4], which states that LexBFS$^+$ sweeps preserve cocomparability orderings.

Theorem 3.5. *[4] Let σ be a cocomparability ordering of $G = (V, E)$. The ordering $\tau = \text{LexBFS}^+(\sigma)$ is a cocomparability ordering of G.*

Another easy consequence of the Flipping Lemma is the following corollary.

Corollary 3.6. *For a non-trivial cocomparability graph G (i.e. $|V(G)| \geq 2$), $\text{LexCycle}(G)$ is necessarily even.*

Proof. If G contains a pair of nonadjacent vertices, then the claim is a trivial consequence of the Flipping Lemma. Otherwise G is a complete graph and $\sigma_2 = \sigma_1^d$ is the cycle of length 2. □

An example of a graph which illustrates that this is not the case for all graphs is the graph G_3 with $\text{LexCycle}(G_3) = 3$ drawn in Figure 2.

If Conjecture 1.2 is true, then Theorems 1.1 and 3.5 together imply that for any starting ordering σ_0, a LexBFS$^+$ multisweep on a cocomparability graph G always ends on a 2-cycle consisting of two cocomparability orderings of G. Therefore, if Conjecture 1.2 is true, we would have the following simple algorithm for getting a transitive orientation of a comparability graph.

Algorithm 2 A *Potential* Simple Transitive Orientation Algorithm

Input: A comparability graph $G = (V, E)$

Output: A comparability order of G

1: Construct G' the complement of G
2: Take an arbitrary order σ_0 on the vertices of G'
3: $\sigma_1 \leftarrow \text{LexBFS}^+(G', \sigma_0)$, $\sigma_2 \leftarrow \text{LexBFS}^+(G', \sigma_1)$
4: $i \leftarrow 2$
5: **while** $\sigma_i \neq \sigma_{i-2}$ **do**
6: $\quad i \leftarrow i + 1$
7: $\quad \sigma_i \leftarrow \text{LexBFS}^+(G', \sigma_{i-1})$
8: **end while**
9: **return** σ_i

4. Domino-free Cocomparability Graphs

In support of Conjecture 1.2, we show in this section that the conjecture holds for the subclass of domino-free cocomparability graphs. This class in particular includes the classes of proper interval, interval and cobipartite graphs, but for these three subclasses we provide independent proofs which imply stronger results. For interval graphs we show that the two orderings of the LexCycle are left endpoint and right endpoint orderings of the *same* interval representation, and that such a cycle is reached in at most n iterations of the multisweep algorithm. Moreover in the case of proper interval graphs, we prove that the cycle is reached in at most 3 iterations and that the 2 cycles are duals one of another. The independent proof for cobipartite graphs is, first of all, interesting for the different flavor of the proof, and secondly it provides an upper bound of $3n$ iterations of multisweep algorithm before reaching the cycle.

4.1. Domino-free cocomparability graphs

Here we prove the more general result of the paper regarding Conjecture 1.2. Recall that a domino is the graph obtained from a cycle of length 6 by adding a diametral chord (see Figure 1).

Theorem 4.1. *Domino-free cocomparability graphs have* LexCycle $= 2$.

Proof. Let $G = (V, E)$ be a domino-free cocomparability graph. Let $\sigma_1, \ldots, \sigma_k$ be a LexBFS$^+$ cycle obtained by a multisweep LexBFS$^+$ process on G, and assume by contradiction that $k > 2$. Recall that by Corollary 3.6, k is even and also because of Theorem 1.1 and Theorem 3.5, we can assume that every σ_i is a cocomparability ordering. For two consecutive orderings of the same parity (index i is considered mod k) :

$$\sigma_i = u_1, u_2, \ldots, u_n \quad \text{and} \quad \sigma_{i+2} = v_1, v_2, \ldots, v_n$$

let diff(i) denote the index of the first (left most) vertex that is different in σ_i, σ_{i+2}:

$$\text{diff}(i) = \min\{j \in [n] \mid u_j \neq v_j\}$$

Now up to "shifting" the start of the cycle, we can assume without loss of generality that diff(1) is minimal amongst all diff(i). Also from now on, in order to use lighter notations, we will write $\prec_i$ instead of $\prec_{\sigma_i}$, and $LMPN(x|_k y)$ instead of $LMPN(x|_{\sigma_k} y)$.

Let then a, b be the first (left most) difference between σ_1 and σ_3. Denoting $\sigma_1 = u_1, u_2, \ldots, u_n$ and $\sigma_3 = v_1, v_2, \ldots, v_n$, and $j = \text{diff}(1)$, we have thus $u_i = v_i, \forall i < j$ and $u_j = a, v_j = b$. Note that this implies in particular $a \prec_1 b$ and $b \prec_3 a$. Furthermore, if we define $S = \{u_1, \ldots, u_{j-1}\} = \{v_1, \ldots, v_{j-1}\}$, then $\sigma_1[S] = \sigma_3[S]$, so at the time a (resp. b) was chosen in σ_1 (resp. σ_3),

b (resp. a) had the same label. Therefore in both cases it means the $^+$ rule was applied to break ties between a and b and so $b \prec_k a$ and $a \prec_2 b$. We thus have :

$$\sigma_k = \ldots b \ldots a \ldots \qquad \sigma_2 = \ldots a \ldots b \ldots$$

$$\sigma_1 = S, a \ldots b \ldots \qquad \sigma_3 = S, b \ldots a \ldots$$

Since $a \prec_1 b$ and $a \prec_2 b$, Lemma 3.3 applies, so we choose vertex c as $c = \text{LMPN}(a|_2 b)$. Using the Flipping Lemma on b and c, we place vertex c in the remaining orderings as follows:

$$\sigma_k = \ldots c \ldots b \ldots a \ldots \qquad \sigma_2 = \ldots c \ldots a \ldots b \ldots$$

$$\sigma_1 = S, a \ldots b \ldots c \ldots \qquad \sigma_3 = S, b \ldots a \ldots \quad \text{and } b \prec_3 c$$

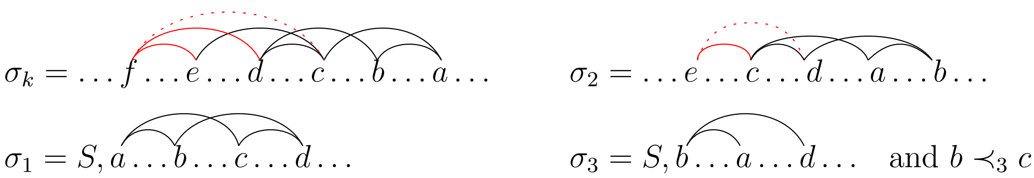
This gives rise to a bad LexBFS triple in σ_k where $c \prec_k b \prec_k a$ and $ca \in E, cb \notin E$. By the LexBFS C_4 Property 3.2, there exists a vertex $d \prec_k c$ such that $d = \text{LMPN}(b|_k a)$ and $dc \in E$. We again use the Flipping Lemma for $ad \notin E$ to place d in the remaining orderings. Note that in σ_2, the Flipping Lemma places $d \prec_2 a$, and by the choice of c as $\text{LMPN}(a|_2 b)$, it follows that no private neighbour of b with respect to a could be placed before c in σ_2. Therefore we can conclude that $c \prec_2 d \prec_2 a$.

$$\sigma_k = \ldots d \ldots c \ldots b \ldots a \ldots \qquad \sigma_2 = \ldots c \ldots d \ldots a \ldots b \ldots$$

$$\sigma_1 = S, a \ldots b \ldots c \ldots \quad \text{and } a \prec_1 d \qquad \sigma_3 = S, b \ldots a \ldots d \ldots \quad \text{and } b \prec_3 c$$

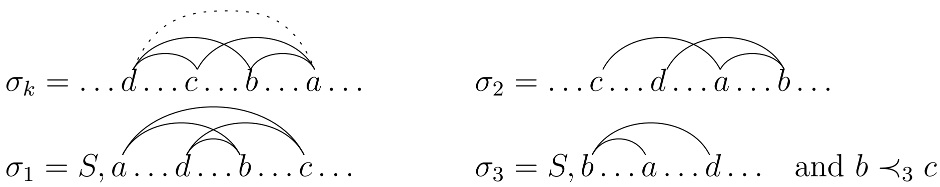
It remains to place d in σ_1 and c in σ_3. We start with vertex d in σ_1. We know that $a \prec_1 d$. This gives rise to three cases: Either **(i)** $c \prec_1 d$, or **(ii)** $a \prec_1 d \prec_1 b$, or **(iii)** $b \prec_1 d \prec_1 c$.

(i). If $c \prec_1 d$ then since $c \prec_2 d$, so we apply Lemma 3.3 and choose a vertex e as $e = \text{LMPN}(c|_2 d)$. This means $ed \notin E$, and since $da \notin E$ and $e \prec_2 d \prec_2 a$, it follows that $ea \notin E$ for otherwise the triple e, d, a would form an umbrella.

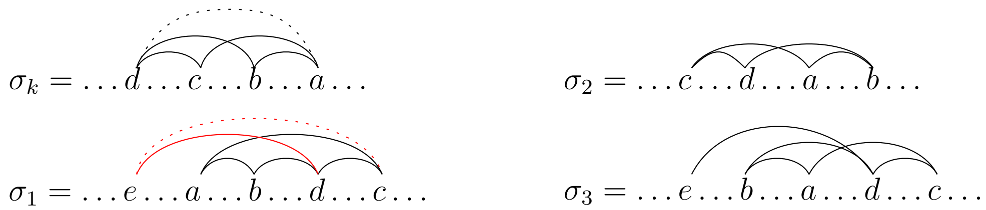


Furthermore, by the choice of vertex c as $\text{LMPN}(a|_2 b)$, and the facts that $e \prec_2 c$ and $ea \notin E$, it follows that $eb \notin E$, otherwise e would be a private neighbour of b with respect to a that is to the left of c in σ_2. Using the Flipping Lemma, we place vertex e in the remaining orderings, and in particular, placing vertex e in σ_k gives rise to a bad LexBFS triple e, d, c. By the LexBFS 4PC and the LexBFS C_4 Property, there must exist a vertex f chosen as $f = \text{LMPN}(d|_k c)$ and $fe \in E$. Using the same argument above, one can show that $fc \notin E$ and $cb \notin E$ implies $fb \notin E$, and given the choice of d in σ_1 and $fb \notin E$, then $fa \notin E$. We, therefore, have the induced domino $abcdef$. A contradiction to G being domino-free.

(ii). If $a \prec_1 d \prec_1 b$, then a, d, b forms a bad LexBFS triple, and thus by Theorem 3.1, choose vertex $e \prec_1 a$ as $e = \text{LMPN}(d|_1 b)$, therefore $eb \notin E$. By the C_4 property (Property 3.2), $ea \in E$. Since $e \prec_1 a$, it follows $e \in S$. But then $ea \in E, eb \notin E$ implies $\text{label}(a) \neq \text{label}(b)$ when a, b were chosen. A contradiction to $S \cap N(a) = S \cap N(b)$.



(iii). We thus must have $b \prec_1 d \prec_1 c$, in which case we still have a bad LexBFS triple given by a, d, c in σ_1. Choose vertex $e \prec_1 a$ as $e = \text{LMPN}(d|_1 c)$ (and remember for later that as explained after Theorem 3.1, e is such that every vertex placed before e is either a common neighbour or a common non-neighbour of c and d). By property 3.2, $ea \in E$, and since $e \prec_1 a$, it follows $e \in S$, and thus $eb \in E$ since $S \cap N(a) = S \cap N(b)$. Since $\sigma_1[S] = \sigma_3[S]$, it follows that e appears in σ_3 in S, and thus e is the $\text{LMPN}(d|_3 c)$ as well. Therefore $d \prec_3 c$. The orderings look as follows:



Consider the ordering of the edge cd in σ_{k-1}. If $d \prec_{k-1} c$, we use the same argument above to exhibit a domino as follows: if $d \prec_{k-1} c$, then $d \prec_{k-1,k} c$, so choose a vertex $p = \text{LMPN}(d|_k c)$. Therefore $pc \notin E$, and since $cb \notin E$ and $p \prec_k c \prec_k b$, it follows that $pb \notin E$ as otherwise we contradict σ_k being a cocomparability ordering. Moreover, given the choice of vertex d in σ_k as the $\text{LMPN}(b|_k a)$ and the fact that $p \prec_k d, pb \notin E$, it follows that $pa \notin E$ as well. We then use the Flipping Lemma to place vertex p in σ_2. This gives rise to a bad LexBFS triple p, c, d in σ_2. Choose vertex $q \prec_2 p$ as $q = \text{LMPN}(c|_2 d)$. Again, one can show that $qa, qb \notin E$, and thus the C_4s in $\{a, b, c, d, p, q\}$ are induced, therefore giving a domino; a contradiction to G being domino-free.

Therefore we must have $c \prec_{k-1} d$. Consider now the first (left most) difference between σ_{k-1} and σ_1. Let S' be the set of initial vertices that is the same in σ_{k-1} and σ_1. By the choice of σ_1 as the start of the cycle $\sigma_1, \sigma_2, \ldots, \sigma_k$, and in particular as the ordering with minimum diff(1), we know that $|S| \leq |S'|$. Since S and S' are both initial segments of σ_1, it follows that $S \subseteq S'$, and the ordering of the vertices in S is the same in S' in σ_1; $\sigma_1[S] \subseteq \sigma_1[S']$. In particular vertex e as constructed above appears in S' as the left most private neighbour of d with respect to c in σ_1, and every vertex before e in σ_{k-1} is either a common neighbour of c and d or a common non-neighbour of c and d. But then d must have been chosen before c, which contradicts $c \prec_{k-1} d$.

Notice that in all cases, we never assumed that $S \neq \emptyset$. The existence of an element in S was always forced by bad LexBFS triples. If S was empty, then case **(i)** would still produce a domino, and cases **(ii)**, **(iii)** would not be possible since $e \in S$ was forced by LexBFS.

To conclude, if G is a domino-free cocomparability graph, then it cannot have $\text{LexCycle}(G) > 2$. □

4.2. Interval graphs

For the special case of interval graphs, we prove a stronger statement about the 2-cycle: it is reached almost as soon as one gets a cocomparability order, and furthermore the two orderings are left and right endpoint ordering of the same interval representation.

Theorem 4.2. *Let G be an interval graph with $|V(G)| > 1$, σ_0 an arbitrary LexBFS cocomparability order of G and $\{\sigma_i,\}_{i\geq 1}$ a sequence of LexBFS$^+$ orderings where $\sigma_i =$ LexBFS$^+(\sigma_{i-1})$. Then the following properties hold :*

- $\sigma_1 = \sigma_3$.
- *There exists an interval representation $\mathcal{R}$ of G such that σ_1 and σ_2 are respectively a left endpoint ordering and a right endpoint ordering of $\mathcal{R}$.*

Before giving the proof of the theorem, let us observe that by Theorem 1.1, in any multisweep LexBFS$^+$ sequence such an order σ_0 is reached in at most n steps, if n is the number of vertices of the graph. Consequently we have that the 2-cycle is reached in at most $n+1$ steps for interval graphs.

Moreover, the second item above implies in particular that σ_1 and σ_2 are I-orderings. This is in fact guaranteed by the following easy lemma.

Lemma 4.3. *Let G be an interval graph, and σ a cocomparability ordering of G. Then $\tau = \text{LexBFS}^+(G, \sigma)$ is an I-ordering of G.*

Proof. Assume by contradiction τ is not an I-ordering. Then there exists a a triple $a \prec_\tau b \prec_\tau c$ where $ac \in E$ and $ab \notin E$. Thus the triple abc forms a bad triple in τ and thus by the LexBFS C_4 property (Property 3.2), there exists a vertex $d \prec_\tau a$ such that d, a, b, c induces a C_4 in G, a contradiction to G being chordal, and thus interval. □

Here is a second lemma that will imply the second item of the Theorem.

Lemma 4.4. *Let G be an interval graph, and σ an I-ordering of G. If $\tau = \text{LexBFS}^+(G, \sigma)$, then there exists an interval representation $\mathcal{R}$ of G such that σ and τ are respectively the left endpoint ordering and the right endpoint ordering of $\mathcal{R}$.*

Proof. Recall that formally σ and τ are bijections from V to $\{1, \dots, n\}$. Define $f_\sigma : V \to \{1, \dots, n\}$ by

$$f_\sigma(v) = \max\{\sigma(w) \mid v \prec_\sigma w \text{ or } w = v\}$$

Informally, $f_\sigma(v)$ is the position in σ of the rightmost neighbour of v to the right of v, or $\sigma(v)$ if there is no such neighbour.

For every vertex v, define the interval $I_v = [\sigma(v), f_\sigma(v)]$ and call $\mathcal{R}$ the resulting collection. Let us prove first that that $\mathcal{R}$ is indeed an interval representation of G. Let u, v be two vertices and assume without loss of generality that $u \prec_\sigma v$ (that is $\sigma(u) < \sigma(v)$). If $uv \in E$, then by definition

$\sigma(v) \leq f_\sigma(u)$ so that I_u and I_v both contain $\sigma(v)$. Conversely if $uv \notin E$, then because σ is an I-ordering, there is no neighbour of u that is placed after v in σ, so we have $f_\sigma(u) < \sigma(v)$, and therefore I_u and I_v are disjoint as required.

By definition σ is a left point ordering of $\mathcal{R}$, so to conclude we have to prove that τ is a right endpoint ordering of $\mathcal{R}$, that is for any vertices u and v, $f_\sigma(u) > f_\sigma(v)$ implies $\tau(u) < \tau(v)$. The inequality on f_σ implies that : either $f_\sigma(u) = \sigma(u)$ and therefore $v \prec_\sigma u$ and $vu \notin E$, or $f_\sigma(u) > \sigma(u)$ and thus there exists w placed after u and v in σ such that $vw \notin E$ and $uw \in E$. But since σ is a cocomparability ordering, the Flipping Lemma 3.4 applies : in the first case we directly get $u \prec_\tau v$ and in the second one we first have $w \prec_\tau v$, which, since τ is an I-ordering, also implies that $u \prec_\tau v$, as required. □

We are now ready for the proof of the main theorem of this subsection.

Proof of Theorem 4.2. Note that the second item follows directly from Lemma 4.3 (applied to $\sigma = \sigma_0$) and Lemma 4.4 (applied to $\sigma = \sigma_1$). Let us thus now prove the first item. Consider the following orderings:

$$\sigma_1 = \text{LexBFS}^+(\sigma_0) \qquad \sigma_2 = \text{LexBFS}^+(\sigma_1) \qquad \sigma_3 = \text{LexBFS}^+(\sigma_2)$$

Suppose, for sake of contradiction, that $\sigma_1 \neq \sigma_3$. Let k denote the index of the first (left most) vertex where σ_1 and σ_3 differ. In particular, let a (resp. b) denote the k^{th} vertex of σ_1 (resp. σ_3). Let S denote the set of vertices preceding a in σ_1 and b in σ_3.

Since the ordering of the vertices of S is the same in both σ_1 and σ_3, and a, b were chosen in different LexBFS orderings, it follows that label$(a) =$ label(b) in both σ_1 and σ_3 when both a and b were being chosen. Therefore, $N(a) \cap S = N(b) \cap S$. So if a were chosen before b in σ_1 then the $^+$ rule must have been used to break ties between label$(a) =$ label(b). This implies $b \prec_0 a$, similarly $a \prec_2 b$. The ordering of the pair a, b is thus as follows:

$$\begin{array}{llcll} \sigma_0: & \ldots b \ldots a \ldots & \qquad & \sigma_2: & \ldots a \ldots b \ldots \\ \sigma_1: & \ldots a \ldots b \ldots & & \sigma_3: & \ldots b \ldots a \ldots \end{array}$$

Using the Flipping Lemma, it is easy to see that $ab \in E$. Since $a \prec_{1,2} b$, we can apply Lemma 3.3 and choose a vertex c as $c = \text{LMPN}(a|_2 b)$. Therefore $c \prec_2 a \prec_2 b$ and $ac \in E, bc \notin E$.

Since σ_0 is a cocomparability order, by Theorem 3.5, $\sigma_1, \sigma_2, \sigma_3$ are cocomparability orderings. Using the Flipping Lemma on the non-edge bc, we have $c \prec_2 b$ implies $c \prec_0 b$. Therefore in σ_0, $c \prec_0 b \prec_0 a$ and $ac \in E, bc \notin E$.

Using the LexBFS 4PC (Theorem 3.1), there exists a vertex d in σ_0 such that $d \prec_0 c \prec_0 b \prec_0 a$ and $db \in E, da \notin E$. By the LexBFS C_4 cocomparability property (Property 3.2), $dc \in E$ and the quadruple $abdc$ forms an induced C_4 in G, thereby contradicting G being an interval graph. □

4.3. Proper Interval Graphs

For proper interval graphs, Corneil proved the following result, which is stronger than Theorem 1.1:

Theorem 4.5. *[2] A graph G is a proper interval graph if and only if the third LexBFS$^+$ sweep on G is a PI-ordering.*

We already know by Theorem 4.2 that the 2 cycle is reached one step after reaching an I-order. For proper interval we prove additionally that the 2 orderings in a 2-cycle are duals one of another.

Theorem 4.6. *Let G be a proper interval graph and σ a PI-ordering of G, then LexBFS$^+(\sigma) = \sigma^d$.*

Proof. Define $\tau = \text{LexBFS}^+(\sigma)$. All we have to prove is that for any vertices $x \prec_\sigma y$ implies $y \prec_\tau x$. For non edges this is exactly Flipping Lemma 3.4, so we can assume that $xy \in E$. Assume by contradiction that $x \prec_\tau y$. Since the pair maintained the same order on consecutive sweeps, we can apply Lemma 3.3 to get a vertex z such that $z \prec_\tau x \prec_\tau y$ and $zx \in E, zy \notin E$. Using the Flipping Lemma, this implies $x \prec_\sigma y \prec_\sigma z$ with $xy, xz \in E$ and $yz \notin E$, which contradicts σ being a PI-ordering. □

Therefore, using Theorem 4.6 and Theorem 4.5, we get Corollary 4.7.

Corollary 4.7. *If G is a proper interval graph with $|V(G)| > 1$, Algorithm 2 stops at $\sigma_5 = \sigma_3$, if not sooner.*

Proof. By Theorem 4.5, we know that σ_3 is a PI-ordering. Using Theorem 4.6, we conclude that Algorithm 2 applied on a PI-ordering computes $\sigma_4 = \sigma_3^d$ and $\sigma_5 = \sigma_4^d = (\sigma_3^d)^d = \sigma_3$. □

4.4. Cobipartite Graphs

In this section we study cobipartite graphs, i.e. graphs whose vertex set can be partitioned into two cliques. These are clearly domino-free (as the complement of a domino contains a triangle), so the fact that such graphs have LexCycle equal to 2 is a consequence of Theorem 4.1. In this section we give a separate proof of this result that we think is interesting for three reasons :

- We prove that the cycle is reached in at most $3n$ sweeps.
- We prove that the cycle is in fact composed of an order and its dual.
- The proof technique sheds light on the link between this problem and the one of doubly lexicographic orderings on rows and columns of matrices.

Let $G = (V = A \cup B, E)$ be a cobipartite graph, where both A and B are cliques. Notice that any ordering σ on V obtained by first placing all the vertices of A in any order followed by the vertices of B in any order is a cocomparability ordering. In particular, such an ordering is precisely how any LexBFS cocomparability ordering of G is constructed, as shown by Lemma 4.9 below. We first show the following easy observation.

Lemma 4.8. *Let G be a cobipartite graph, and let σ be a cocomparability ordering of G. In any triple of the form $a \prec_\sigma b \prec_\sigma c$, either $ab \in E$ or $bc \in E$.*

Proof. Suppose otherwise, then if $ac \in E$, we contradict σ being a cocomparability ordering, and if $ac \notin E$, then the triple abc forms a stable set of size 3, which is impossible since G is cobipartite. □

Lemma 4.9. *Let G be a cobipartite graph, and let $\sigma = x_1, x_2, \dots x_n$ be a LexBFS cocomparability ordering of G. There exists $i \in [n]$ such that $\{x_1, \dots, x_i\}$ and $\{x_{i+1}, \dots, x_n\}$ are both cliques.*

Proof. Let i be the largest index in σ such that $\{x_1, \dots, x_i\}$ is a clique. Suppose $\{x_{i+1}, \dots, x_n\}$ is not a clique, and consider a pair of vertices x_j, x_k where $x_j x_k \notin E$ and $i + 1 \leq j < k$. By the choice of i, vertex x_{i+1} is not universal to $\{x_1, \dots, x_i\}$. Since σ is a LexBFS ordering, vertex x_j is also not universal to $\{x_1, \dots, x_i\}$ for otherwise label(x_j) would be lexicographically greater than label(x_{i+1}) implying $j < i + 1$ - unless $i + 1 = j$, in which case x_j is x_{i+1} and we just showed that x_{i+1} is not universal to $\{x_1, \dots, x_i\}$, thus x_j is also not universal to $\{x_1, \dots, x_i\}$. Let $x_p \in \{x_1, \dots, x_i\}$ be a vertex not adjacent to x_j. We thus have $x_p \prec_\sigma x_j \prec_\sigma x_k$ and both $x_p x_j, x_j x_k \notin E$. A contradiction to Lemma 4.8 above. □

Since cobipartite graphs are cocomparability graphs, by Theorem 1.1, after a certain number $t \leq n$ of iterations, a series of LexBFS$^+$ sweeps yields a cocomparability ordering σ_t. By Lemma 4.9, this ordering consists of the vertices of one clique A followed by another clique B.

Assume $a_1, \ldots, a_p, b_q, \ldots, b_1$ is the ordering of σ_t (the reason why the indices of B are reversed will be clear soon). Consider the $p \times q$ matrix M defined as follows:

$$M_{i,j} = \begin{cases} 1 \text{ if } a_i b_j \in E \\ 0 \text{ otherwise} \end{cases}$$

(All through this section, and for any matrix A, we will denote by $A_{i,j}$ the coefficient of A on row i and column j.)

The easy but crucial property that follows from the definition of LexBFS is the following: the columns of this matrix M are sorted lexicographically in increasing order (for any pair of vectors of the same length X and Y, lexicographic order is defined by $X <_{lex} Y$ if the least integer k for which $X_k \neq Y_k$ satisfies $X_k < Y_k$).

Consider $\sigma_{t+1} = \text{LexBFS}^+(\sigma_t)$, and notice that σ_{t+1} begins with the vertices of B in the ordering $b_1, b_2, \ldots, b_q$ followed by the vertices of A which are sorted exactly by sorting the corresponding rows of M lexicographically in non-decreasing order (the first vertex to appear after b_q being the maximal row, that is the one we put at the bottom of the matrix). But then to obtain σ_{t+2} we just need to sort the columns lexicographically, and so on.

Therefore to prove that LexCycle $= 2$ for cobipartite graphs, it suffices to show that this process must converge to a fixed point: that is, after some number of steps, we get a matrix such that both rows and columns are sorted lexicographically, which implies we have reached a 2 cycle. This is guaranteed by the following Proposition (which we state for $0-1$ matrices, but the proof works identically for any integer valued matrix).

Proposition 4.10. *Let M be a $p \times q$ matrix with $\{0, 1\}$ entries. Define two sequences of matrices $(R^{(t)})_{t \geq 0}$ and $(C^{(t)})_{t \geq 1}$ as follows:*

- $R^{(0)} = M$
- *For $t \geq 1$, $C^{(t)}$ is obtained by sorting the columns of $R^{(t-1)}$ in non-decreasing lexicographical order.*
- *For $t \geq 1$, $R^{(t)}$ is obtained by sorting the rows of $C^{(t)}$ in non-decreasing lexicographical order.*

Then there exists $k \leq q$ for which $R^{(k)} = C^{(k)}$

Note that the conclusion in fact implies that both sequences are constant from the index k since this implies that $R^{(k)}$ has both its rows and columns sorted lexicographically. This proposition is reminiscent of the doubly lexical ordering of $\{0,1\}$ matrices studied by Lubiw in [21]. What we prove implies that in order to obtain this doubly lexical ordering, one can do in fact a sequence of at most n LexBFS, giving thus a $O(nm)$ time, if m denotes the number of non zero entries. Better algorithms for this problem are already known. For instance, in [25], Spinrad gave an $O(n^2)$ time for dense matrices.

Proof of Proposition 4.10. We rely on the following claim.

Claim : For every t, the t first columns of $R^{(t)}$ are sorted in non decreasing lexicographic order, and are smaller than the $q-t$ last columns of the matrix.

This indeed implies the desired result, as for $t = q$ we have that the whole matrix $R^{(q)}$ is doubly lexicographic (rows by construction and columns follow from the claim), which proves our Proposition.

We prove the Claim by induction on t. This is obvious for $t = 0$, so let us assume that $t \geq 1$ and that the property is true for $t-1$. The induction hypothesis implies that in $C^{(t)}$, the $(t-1)$-first columns are identical to those of $R^{(t-1)}$. Since the rows of $R^{(t-1)}$ were sorted (by definition), we have that the matrix resulting from $C^{(t)}$ by looking just at the first $(t-1)$-columns is sorted both for rows and columns. Therefore in $R^{(t)}$, these same entries will also be identical and so the $(t-1)$ first columns of $R^{(t)}$ are sorted.

Assume by contradiction that for some $p \leq t$ and $q \geq t$, the q-th column of $R^{(t)}$ is strictly smaller than the p-th column. Let then i be the smallest integer such that $R^{(t)}_{i,p} \neq R^{(t)}_{i,q}$, and thus $R^{(t)}_{i,p} = 1$ and $R^{(t)}_{i,q} = 0$. Since $R^{(t)}$ was obtained from $C^{(t)}$ by a reordering of its rows, there exists i' such that $C^{(t)}_{i',p} = 1$ and $C^{(t)}_{i',q} = 0$ – see Figure 5.

Since the columns in $C^{(t)}$ are sorted, we deduce that column p in $C^{(t)}$ is lexicographically smaller than column q in $C^{(t)}$. Since $C^{(t)}_{i',p} = 1$ and $C^{(t)}_{i',q} = 0$, there must exist $j' < i'$ such that $C^{(t)}_{j',p} = 0$ and $C^{(t)}_{j',q} = 1$ as illustrated below.

Row j' must appear somewhere in $R^{(t)}$, which was obtained by sorting rows of $C^{(t)}$. Recall, however, that first $(t-1)$ columns of $C^{(t)}$ have their rows sorted. And thus in particular, the first $(t-1)$ elements of the j'^{th} row of $C^{(t)}$ are the same or lexicographically smaller than the first $(t-1)$ elements of the i'^{th} row of $C^{(t)}$. This means that sorting the rows of $C^{(t)}$ puts row j' above row i'. Since row i' lands at row i in $R^{(t)}$, row j' must land above row i in $R^{(t)}$. But all rows above row i in $R^{(t)}$ have identical element in the p^{th} and q^{th} columns by the minimality of i, while j' does not, a contradiction.

This concludes the induction. □

We conclude with the following corollary:

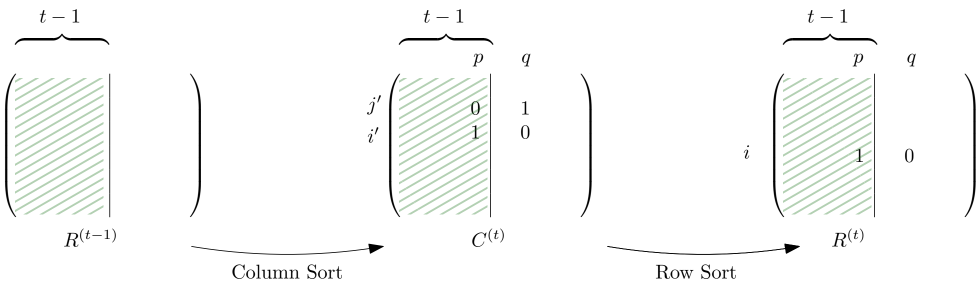


Figure 5: The matrices in Proposition 4.10

Corollary 4.11. *Cobipartite graphs have* LexCycle = 2*, this cycle is reached in fewer than* $3n$ LexBFS$^+$ *sweeps, and the two orderings that witness* LexCycle = 2 *are duals of one another.*

Proof. As mentioned before, Theorem 1.1 implies that in at most n sweeps one gets a cocomparability order. From this point we know that we can view the successive sweeps through the incidence matrix of the edges between the two cliques. Each sequence of two consecutive sweeps corresponds to sorting the columns and then the rows of the matrix, and therefore by Proposition 4.10 above we get that this converges in less that $2n$ more sweeps to a fixed matrix. Now this means that we have reached a cycle of length 2 and that indeed the two orderings are duals one of another. □

5. Conclusion & Perspectives

In this paper, we study a new graph parameter, LexCycle, which measures the maximum length of a cycle of LexBFS$^+$ sweeps. We believe it reflects some measure of linearity structure of a class of graphs: if the class has some strong linear structure, LexCycle should be small for this class. For example, interval graphs have a clear definition of linearity: they are the graphs that arise from the intersection of intervals on the real line. Cocomparability graphs, and more generally AT-free graphs, also have a notion of linearity. Every AT-free graph, and thus every cocomparability graph, admits a dominating diametral path [6]. This dominating diametral path has been known in the literature as the *spine* of the graph [6], because it opens the graph in a linear fashion, where vertices not on path hang at distance one. We have no example of an asteroidal triple-free graph whose LexCycle is larger than 2, so it may be the case that Conjecture 1.2 is even true for AT-free graphs (which we note have asteroidal number 2).

Note however that as shown in this paper, a stronger conjecture stating LexCycle is bounded above by the asteroidal number is false (this was the question of Stacho for which we provided a counterexample in Section 2). Our construction suggests that perhaps LexCycle cannot be bounded by any polynomial function on the number of vertices.

Towards proving Conjecture 1.2 about cocomparability graphs, we showed that domino-free cocomparability graphs (which contain cographs, interval graphs, cobipartite graphs) all have LexCycle = 2. Next, we motivate the choice of the domino as a forbidden structure and a potential direction to prove the conjecture for cocomparability graphs.

Define a k**-ladder** to be an induced graph of k *chained* C_4s. More precisely, a ladder is a graph $H = (V_H, E_H)$ where $V_H = \{x_0, x_1, x_2, \ldots, x_k, y_0, y_1, \ldots, y_k\}$ and $E_H = \{(x_i, y_i), (x_i, x_{i+1}), (y_i, y_{i+1}) : \forall i, 0 \leq i \leq k-1\} \cup \{(x_k, y_k)\}$, as illustrated in Figure 6.

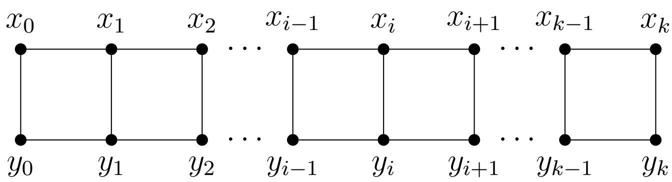


Figure 6: A k-ladder.

Observe that interval graphs are equivalent to 1-ladder-free cocomparability graphs, and domino-free graphs are precisely 2-ladder-free graphs. Therefore we believe that the study of k-ladder-free cocomparability graphs is a good strategy towards proving LexCycle=2 for cocomparability graphs.

Recently there has been progress on other subclasses of cocomparability graphs: in [10], the authors showed that $\overline{P_2 \cup P_3}$-free cocomparability graphs, and thus diamond-free cocomparability graphs as well as cocomparability graphs with girth 4, have LexCycle = 2.

Outside cocomparability graphs, we wonder if the conjecture holds for split graphs as well. And, although we haven't been able to prove the conjecture for trees, we strongly believe that it holds for trees using BFS only – the lack of cycles on trees implies that every LexBFS is a BFS ordering.

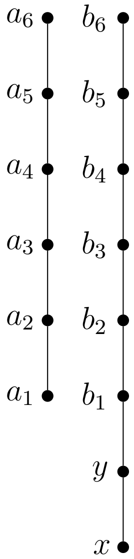
A word on runtime for arbitrary cocomparability graphs: Although the conjecture is still open for cocomparability graphs, experimentally we observed that the convergence often happens relatively quickly, but not always, as shown by the sequence of graphs $\{G_n\}_{n \geq 2}$ presented below. This graph family, experimentally, takes $O(n)$ LexBFS$^+$ sweeps before converging. We describe an example in the family in terms of its complement since it is easier to picture, and the LexBFS traversals of the complement are easier to parse. Let $G_n = (V = A \cup B, E)$ be a *comparability* graph on $2n+2$ vertices, where both

A and B are paths, i.e. $A = a_1, a_2, \ldots, a_n, B = x, y, b_1, b_2, \ldots, b_n$, and the only edges in E are of the form $E = \{(a_i a_{i+1}) : i \in [n-1]\} \cup \{(xy), (yb_1)\} \cup \{(b_j b_{j+1}) : j \in [n-1]\}$. The initial comparability ordering is constructed by collecting the odd indexed vertices first, then the even indexed ones as follows:

- Initially we start τ with x, a_1.
- In general, if the last element in τ is a_i and i is odd, while i is in a valid range, append b_i, b_{i+2}, a_{i+2} to τ and repeat.
- If n is even, append b_n, a_n to τ, otherwise append a_{n-1}, b_{n-1} to τ.
- Again while i is in a valid range, we append the even indexed vertices $a_i, b_i, b_{i-2}, a_{i-2}$ to τ.
- Append y to τ.

The ordering τ as constructed is a transitive orientation of the graph, and thus is a cocomparability ordering in the complement. We perform a series of LexBFS$^+$ sweeps where σ_1 = LexBFS$^+(\tau)$ in the complement, i.e. the cocomparability graph.

Every subsequent $^+$ sweep will proceed to "gather" the elements of A close to each other, resulting in an ordering that once it moves to path A remains in A until all its elements have been visited. An intuitive way to see why this must happen is to notice in the complement, the vertices of A are universal to B and thus must have a strong pull. Experimentally, this 2-paths graph family takes $O(n)$ LexBFS$^+$ sweeps before converging. Figure 7 below is an example for $n = 6$.



$$
\begin{aligned}
\tau & \quad = x, a_1, b_1, b_3, a_3, a_5, b_5, b_6, a_6, a_4, b_4, b_2, a_2, y \\
\sigma_1 = \text{LexBFS}^+(\tau) & \quad = y, a_2, b_2, b_4, a_4, a_6, b_6, a_5, b_5, b_3, a_1, a_3, x, b_1 \\
\sigma_2 = \text{LexBFS}^+(\sigma_1) & \quad = b_1, x, a_1, a_3, b_3, b_5, a_5, a_6, b_6, b_4, a_4, a_2, b_2, y \\
\sigma_3 = \text{LexBFS}^+(\sigma_2) & \quad = y, b_2, a_2, a_4, b_4, b_6, a_6, b_5, a_1, a_1, a_3, b_3, x, b_1 \\
\sigma_4 = \text{LexBFS}^+(\sigma_3) & \quad = b_1, x, b_3, a_3, a_1, a_5, b_5, b_6, a_6, a_4, a_2, b_4, b_2, y \\
\sigma_5 = \text{LexBFS}^+(\sigma_4) & \quad = y, b_2, b_4, a_2, a_4, a_6, b_6, a_5, a_1, a_3, b_5, b_3, x, b_1 \\
\sigma_6 = \text{LexBFS}^+(\sigma_5) & \quad = b_1, x, b_3, b_5, a_3, a_1, a_5, a_6, a_4, a_2, b_6, b_4, b_2, y \\
\sigma_7 = \text{LexBFS}^+(\sigma_6) & \quad = y, b_2, b_4, b_6, a_2, a_4, a_6, a_5, a_1, a_3, b_5, b_3, x, b_1 \\
\sigma_8 = \text{LexBFS}^+(\sigma_7) & \quad = b_1, x, b_3, b_5, a_3, a_1, a_5, a_6, a_4, a_2, b_6, b_4, b_2, y = \sigma_6
\end{aligned}
$$

Figure 7: G_6, A comparability graph; τ a cocomparability ordering of the complement of G_6 and a series of LexBFS$^+$ of the corresponding cocomparability graph.

Other Graph Searches: One could raise a similar cycle question for different graph searches; in particular, *Lexicographic Depth First Search* (LexDFS). LexDFS was introduced in [5] and is a graph search that extends DFS is a similar way to how LexBFS extends BFS - see Algorithm 3.

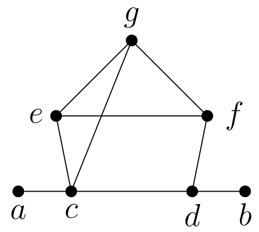
Algorithm 3 LexDFS

Input: A graph $G = (V, E)$ and a start vertex s

Output: An ordering σ of V

```
assign the label ϵ to all vertices, and label(s) ← {0}
for i ← 1 to n do
    pick an unnumbered vertex v with lexicographically largest label
    σ(v) ← i                                  ▷ v is assigned the number i
    foreach unnumbered vertex w adjacent to v do
        prepend i to label(w)
    end for
end for
```

LexDFS has led to a number of linear time algorithms on cocomparability graphs, including maximum independent set and Hamilton path [3, 4, 18]. In fact, these recent results have shown just how powerful combining LexDFS and cocomparability orderings is. It is therefore natural to ask whether a sequence of LexDFS orderings on cocomparability graphs reaches a cycle with nice properties. Unfortunately, this is not the case as shown by the



$$
\begin{aligned}
\sigma_1 &= \text{LexDFS}(G) &&= a, c, d, b, f, g, e \\
\sigma_2 &= \text{LexDFS}^+(\sigma_1) &&= e, g, f, d, c, a, b \\
\sigma_3 &= \text{LexDFS}^+(\sigma_2) &&= b, d, c, a, g, e, f \\
\sigma_4 &= \text{LexDFS}^+(\sigma_3) &&= f, e, g, c, d, b, a \\
\sigma_5 &= \text{LexDFS}^+(\sigma_4) &&= a, c, d, b, f, g, e = \sigma_1
\end{aligned}
$$

Figure 8: A sequence of LexDFS$^+$ orderings on a cocomparability graph, that cycles after 5 iterations, and none of the orderings is a cocomparability order.

example in Figure 8, where G is a cocomparability graph as witnessed by the following cocomparability ordering $\tau = a, c, e, f, g, d, b$, however doing a sequence of LexDFS$^+$ on G cycles before we reach a cocomparability ordering, and the cycle has size four.

Acknowledgements: The authors wish to warmly thank the anonymous referees for their very helpful comments that helped us correct flaws in our proofs. They also thank Jérémie Dusart and Derek Corneil for helpful discussions on this subject.

References

[1] Bandelt, Hans-Jürgen and Mulder, Henry Martyn: Distance-hereditary graphs. Journal of Combinatorial Theory, Series B 41(2), 182–208 (1986)

[2] Corneil, Derek G.: A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs. Discrete Applied Mathematics 138, 371–379 (2004)

[3] Corneil, Derek G. and Dalton, Barnaby and Habib, Michel: LDFS-based certifying algorithm for the minimum path cover problem on cocomparability graphs. SIAM Journal on Comput. 42, 792–807 (2013)

[4] Corneil, Derek G. and Dusart, Jérémie and Habib, Michel and Kőhler, Ekkehard: On the power of graph searching for cocomparability graphs. SIAM J. Discrete Math. 30, 569–591 (2016)

[5] Corneil, Derek G. and Krueger, Richard: A unified view of graph searching. SIAM J. Disc. Math. 22, 1259–1276 (2008)

[6] Corneil, Derek G. and Stephan Olariu and Lorna Stewart: Linear Time Algorithms for Dominating Pairs in Asteroidal Triple-free Graphs. SIAM J. Comput. 28, 1284–1297 (1999)

[7] Corneil, Derek G. and Stephan Olariu and Lorna Stewart: The LBFS Structure and Recognition of Interval Graphs. SIAM J. Discrete Math. 23, 1905–1953 (2009)

[8] Dragan, F. Feodor and Falk, Nicolai and Brandstädt, Andreas: LexBFS-orderings and powers of graphs. Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science, 166–180 (1996)

[9] Dusart, Jérémie and Habib, Michel: A new LBFS-based algorithm for cocomparability graph recognition. Discrete Applied Mathematics 216, 149–161 (2017).

[10] Gao, Xiao-Lu and Xu, Shou-Jun: The LexCycle on $\overline{P_2 \cup P_3}$−free Cocomparability Graphs https://arxiv.org/abs/1904.08076 (2019)

[11] Gilmore, Paul C. and Hoffman, Alan J.: A characterization of comparability graphs and of interval graphs Canadian Journal of Mathematics. 16, 539–548 (1964)

[12] Golumbic, Martin C. and Monma, Clyde L. and Trotter, William T. Jr: Tolerance graphs. Discrete Applied Math. 9, 157–170 (1984)

[13] Habib, Michel and Mouatadid, Lalla: Maximum Induced Matching Algorithms via Vertex Ordering Characterizations: Algorithmica. 1432-0541, 1–19 (2019).

[14] Habib, Michel and McConnell, Ross M. and Paul, Christophe and Viennot, Laurent: Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. Theor. Comput. Sci. 234, 59–84 (2000)

[15] Jordan, Camille: Sur les assemblages de lignes. Journal für reine und angewandte Mathematik 70, 185–190 (1869)

[16] Kratsch, Dieter and Stewart, Lorna: Domination on Cocomparability Graphs. SIAM J. Discrete Math. 6, 400-417 (1993)

[17] Kratsch, Dieter and McConnell, Ross M. and Mehlhorn, Kurt and Sprinrad, Jeremy P.: Certifying algorithms for recognizing interval graphs and permutation graphs. SIAM Journal on Computing. 36, 326–353 (2006)

[18] Köhler, Ekkehard and Mouatadid, Lalla: Linear Time LexDFS on Cocomparability Graphs. Proceedings of the Fourteenth Symposium and Workshop on Algorithm Theory, 319–330 (2014)

[19] Köhler, Ekkehard and Mouatadid, Lalla: A linear time algorithm to compute a maximum weighted independent set on cocomparability graphs. Information Processing Letters 116, 391–395 (2016)

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

[21] Lubiw, Anna.: Doubly lexical orderings of matrices. SIAM Journal on Computing 16, 854–879 (1987)

[22] Robert Paige and Robert Endre Tarjan. Three Partition Refinement Algorithms. SIAM Journal on Computing 16, 973–989 (1987)

[23] Rose, Donald J. and Tarjan, Robert E. and Lueker, George S.: Algorithmic Aspects of Vertex Elimination on Graphs. SIAM J. Comput. 5, 266–283 (1976)

[24] Simon, Klaus: A New Simple Linear Algorithm to Recognize Interval Graphs. Workshop on Computational Geometry. 289–308 (1991)

[25] Spinrad, Jeremy P. Doubly Lexical Ordering of Dense 0–1 Matrices. Information Processing Letters. 45, 229–235 (1993)

[26] Roberts, Fred S.: Indifference Graphs. Proof Techniques in Graph Theory. 139–146 (1969)

[27] Stacho, Juraj: Private communication. (2014)