HABILITATION À DIRIGER DES RECHERCHES

Spécialité Informatique

École doctorale n°386 - Science Mathématiques Paris-Centre

UNIVERSITÉ PARIS DIDEROT (PARIS VII)

ABOUT SOME HEREDITARY CLASSES OF GRAPHS : ALGORITHMS - STRUCTURE - COLORATION

Mémoire d'habilitation à diriger des recherches présenté et soutenu publiquement par

Pierre Charbit

le 16 Mai 2019

devant le jury composé de MM.

Rapporteurs

Christophe PAUL,	CNRS - Université de Montpellier
Alexander SCOTT,	Oxford University
Laurent VIENNOT,	INRIA - Université Paris VII

Examinateurs

Frédéric HAVET,	CNRS - Sophia Antipolis
Claire MATHIEU,	CNRS - Université Paris VII
Jean-Sébastien SERENI,	CNRS - Université de Strasbourg

Président du Jury

M. Pierre FRAIGNIAUD,	CNRS - Université Paris VII

Contents

Remerciements

Je tiens à remercier tout d'abord Christophe Paul, Alex Scott et Laurent Viennot d'voir accepté d'être rapporteurs pour ce document. Je remercie aussi Pierre Fraigniaud d'avoir accepté de présider ce jury et Frédéric Havet, Claire Mathieu et Jean Sébastien Séreni d'avoir accepté d'en faire partie. Je suis très honoré de pouvoir présenter mes travaux devant tous ce panel de chercheur.e.s de renom.

Ce document expose des travaux effectués au sein du laboratoire LIAFA (désormais IRIF) dans lequel je suis arrivé en Septembre 2006. Je veux ici souligner l'ambiance particulièrement agréable, saine et stimulante que j'y ai trouvé dès mon arrivée et qui persiste jusqu'à ce jour. Je remercie tout particulièrement Michel Habib qui m'a accueilli avec la chaleur qui est la sienne dans l'équipe Graphes et Algorithmes qu'il venait de monter au LIAFA. Les rencontres que j'y ai faites ont beaucoup influencé les thèmes de recherche sur lesquels j'ai par la suite travaillé, et j'ai eu énormément de plaisir avec à échanger avec lui, Fabien de Montgolfier, Mathieu Raffinot et Laurent Viennot dès mes premières années ici. J'ai été très heureux aussi de retrouver et de côtoyer Nicolas Trotignon et Jean Sébastien Sereni, durant les trop courtes années qu'is ont passé au laboratoire. Notre équipe a vu passer beaucoup de doctorant.e.s, post doctorant.e.s avec qui j'ai pu travailler, transmettre et apprendre, je pense notamment à Vincent Limouzy, Mickaël Rao, Pierre Aboulker, Théophile Trunk, Juraj Stacho, Yilma Zelealem, Antoine Mamcarz, Jérmie Dusart, Lalla Mouatadid, Lucas Hosseini, Carl Feghali ou Valia Mitsou. L'arrivée de Reza Naserasr dans le labo (et mon bureau) depuis quatre ans a donné un nouveau souffle à notre équipe, et je suis très heureux de pouvoir travailler avec lui. Je ne peux pas ne pas mentionner Christophe Prieur qui a été un compagnon plus qu'agréable pendant les nombreuses années durant lesquelles nous avons partagé un bureau. Merci enfin à tous les collègues enseignant.e.s de l'UFR qui fournissent un travail considérable et qui collectivement arrivent malgré tout à maintenir une dynamique de recherche stimulante. Merci aussi au membres du personnel administratif pour leur soutien. Un seul admin vous manque et tout est paralysé, pour mal paraphraser un adage célèbre mais pour dire une évidence trop peu souvent rappelée (et dont ce laboratoire peut témoigner..). Merci donc à Noëlle Delgado, Odile Ainardi, Laifa Ahmadi, Houy Kuoy, Etienne Mallet, Omur Avci, Dieneba Camara, Eva Ryckelynck et Florence Thépenier.

Je tiens aussi à remercier l'équipe MC2 du LIP à l'ENS Lyon au sein de laquelle j'ai passé une année de délégation CNRS très stimulante entre Septembre 2017 et Septembre 2018. Evidemment je pense en premier lieu à Stéphan Thomassé avec qui j'ai eu bonheur à reformer une complicité scientifique qui datait de mes désormais lointaines années lyonnaises. J'y ai retrouvé Nicolas Trotignon et ai pu rencontrer les jeunes chercheurs brillants que sont William Lochet, Rémi Watrigant et Edouard Bonnet. Enfin la recherche se nourrit de nombreuses interactions et je veux mentionner d'autres chercheurs et chercheuses avec qui j'ai aimé échanger au cours de cette décennie, je pense en particulier à Marthe Bonamy, Nicolas Bousquet, Louis Esperet, Gwenaël Joret, Aurélie Lagoutte, Patrice Ossona de Mendez et Kristina Vušković. Je tiens aussi à remercier Maria Chudnovsky, Paul Seymour et Bruce Reed pour avoir organisé année après année des workshops fantastiques au cours lesquels j'ai pu apprendre énormément de ce que je connais en théorie des graphes. Merci enfin encore à Adrian Bondy sans la rencontre de qui, à Lyon il y a presque 20 ans, je ne me serai probablement jamais engagé dans cette thématique passionnante.

Merci aussi à mes amis et amies, je suis chanceux d'en avoir trop pour pouvoir tous les remercier ici, merci à mes parents, à Manu et Missy pour leur soutien et leur amour. Merci à ceux qui auront voulu me faire le plaisir d'être là aujourd'hui pour m'entendre parler de choses invraisemblables.

Merci enfin infiniment à Julie pour sa présence si importante dans ma vie, à Lucien pour sa parfois bruyante mais irrésistible joie de vivre et à Joanne d'avoir été une merveille de douceur pendant les quatre premières semaines de sa vie, me laissant le temps nécessaire à la préparation de la soutenance d'aujourd'hui.

Chapter 1

Introduction

1.1 Hereditary classes of graphs

As a mathematical field, Graph Theory has for object the understanding of some abstract objects - here graphs - through the study of some of their properties, chosen because they seem natural with respect to other mathematical preexisting constructions, or for aesthetic reasons, or because they are relevant to applications. Even if the following adjectives are not formally precise and can overlap, one can have in mind properties that are topological, (e.g. the graph is planar : it can be drawn in the euclidean plane without crossing of edges), or combinatorial (e.g. one can label its vertices with 3 labels such that adjacent vertices get different labels), or structural (e.g. the graph does not contain 3 pairwise adjacent vertices). For some natural reasons, many properties that are studied are what we call *hereditary* : if a graph satisfies it, then any subgraph of it (in a sense that has to be specified) also satisfies the property. This is the case of the three examples given above if subgraph is understood as the result of deletion of some vertices and edges.

One notable kind of property definition is the third example given above : ”the class of graphs that do not contain 3 pairwise adjacent vertices”, or in different words, graphs that do not contain a triangle. Indeed any class defined this way - by a list of forbidden substructures - is obviously hereditary. Conversely, any hereditary class $\mathcal{C}$ can be characterized as ”the class of graphs that do not contain any graph $\mathcal{F}$” for some family $\mathcal{F}$. It is obvious if one takes $\mathcal{F}$ to be the complement of $\mathcal{C}$, but the smallest such family is unique and is easily seen to be the set of graphs that are not in $\mathcal{C}$, and minimal for this property, with respect the the chosen subgraph relation. For the usual subgraph relation this minimal set of obstructions is not necessarily finite (think of acyclic graphs - forests - being defined minimally by forbidding the collection of all cycles). Amongst the nicest theorems in Graph Theory are the one that manage to establish an equivalence between two different types of properties. The classical Kuratovski's theorem characterising planar graphs by two forbidden substructures is of this kind since it relates a topological notion with a structural one, and the celebrated Strong Perfect Graph Theorem (which we will explain a bit later) is another, by linking a combinatorial global property (chromatic number) and a structural one (forbidden subgraphs).

As a computer science topic, Graph Theory is concerned with algorithms. Many interesting decision problems that we encounter in Graph Theory are difficult to solve. Some problems can be solved fast (in polynomial time) and it is a challenge to devise as efficient algorithm as possible, as for example algorithms that are linear in the size of the instance. For those that are not believed to be polynomial (that is, unless P=NP), there is a whole area of exploration and improvement, as approximation algorithms or the theory of FPT-algorithms. For some others the status is simply unknown and it is very stimulating to wonder whether there might exist a polynomial time algorithm to be discovered because finding one usually means that we have a much better understanding of the question. For every possible case, one angle of study is to think about the problem when the instance is restricted to a particular class of graphs, and in particular hereditary classes, as forbidding substructures might of course help devise efficient algorithm. There are many classes for which questions like computing the independence number or the chromatic number are very active subjects and where one can sometimes prove beautiful polynomial time algorithms whereas in the general case these are NP-complete problems.

1.2 An Insightful Example : Chordal Graphs

A graph is *chordal* if it does not contain any *hole*, that is an induced cycle of length at least 4. The class of chordal graphs is very classical and has been intensively studied. A perfect elimination ordering of a graph is an ordering of the vertices where for every vertex v, the set of neighbours of v that precede v in the order induce a complete graph. Here are four properties of chordal graphs, that we could like to generalise to some other hereditary classes.

(1) Decomposition theorem: If G is a connected chordal graph, then either G is complete or it contains a complete subgraph whose removal disconnects the graph into two smaller pieces.

(2) Elimination ordering characterisation: G is chordal if and only if G has a perfect elimination ordering.

(3) Algorithmic properties: Chordal graphs can be recognised in linear time, and optimisation problems such as finding a maximum clique, or a stable set, or the chromatic number can be solved in linear time for chordal graphs.

(4) Colouring Properties If G is chordal then G is *perfect* : its chromatic number is equal to the size of the maximum complete subgraph it contains

These are not orthogonal categories or things to put at the same level - it is usually the case that decomposition theorems and ordering characterisations are proved in order to design better algorithms or prove combinatorial theorems, such as theorems on colouring. These four aspects of chordal graphs will nevertheless serve as a guideline to the work I want presented in this document.

1.3 Structure of the Document

The end of the current chapter will contain some general definitions and notations used throughout this manuscript. Several hereditary properties will reappear in various places in this document so I choose to regroup in Chapter 2 some well known facts about these classical classes or generic types of hereditary properties. Each of the following chapters will present a short survey and results on a theme described above for chordal graphs (the references are the article authored by myself whose results are cited in the chapter).

- Chapter 3 about results on graph decompositions - [CDR12] [Cha+12].
- Chapter 4 about results related to linear orderings and searches, and in particular a search algorithm called LexBFS - [Abo+15] [Cha+17] [CHM14]. [Cha+16].
- Chapter 5 about the computation of maximum clique (or maximum independent set) in two settings, namely H-free graphs for small or specific H, and Ball Graphs, which are intersection graphs of balls in $\mathbb{R}^d$. In this Chapter we will be interested in complexity issues beyond the polynomial world : approximation algorithms and FPT algorithms - [Bon+18a] [Bon+18b].
- Chapter 6 about results on the chromatic number of some hereditary classes, and more specifically the relation between clique number and chromatic number - [Abo+18] [BCT14][Cha+16][ACN].

1.4 Definitions, Notations, Standard Vocabulary

Let us put here the standard definitions and notations that we will use. A *unoriented graph* G (or more simply, a *graph*) is a pair of finite sets $(V(G), E(G))$ where $E(G)$ consists of (non-ordered) pairs of elements of $V(G)$. The elements of $V(G)$ are called *vertices* of G and those of $E(G)$ *edges* of G. If no confusion is possible, we will note V and E instead of $V(G)$ and $E(G)$. We will also use the simplified notation xy or yx for the edge $\{x, y\}$. An edge of the type xx is called a *loop* of G.

If $e = xy$ is an edge of G, we say that x and y are *neighbours* in G and that they are the *endpoints* of e. The *neighbourhood* of a vertex x in a graph G is the set of his neighbours, and we denote it by $N_G(x)$ or $N(x)$ if no confusion is possible. The *degree* of x in G is the cardinality of $N_G(x)$, et is denoted by $d_G(x)$, or $d(x)$. For a subset X of V we define $N_G(X)$ as the union of the neighbourhoods of the vertices in X.

The *complement* of a graph G, denoted $\overline{G}$, is the graph $(V(G), \overline{E}(G))$, where $\overline{E}(G) = \{xy,\, x \neq y$ and $xy \notin E(G)\}$.

As it is the usage in graph theory, in the rest of the document and if no confusion occurs, the letters n and m will always denote respectively the number of vertices and the number of edges of the graph.

Substructures A *subgraph* H of a graph G is a graph that satisfies $V(H) \subset V(G)$ and $E(H) \subset E(G)$. In that case we say that G contains H as a subgraph. If in addition for every pair x, y of vertices in H, xy is an edge in H if and only if xy is an edge in G, we say that H is an *induced subgraph* of G. If $V(H) = V(G)$, the subgraph H is *spanning*.

If $X \subset V$, we denote by $G[X]$ the induced subgraph of G which vertex set is X, by $G \setminus X$ the induced subgraph which vertex set is $V \setminus X$. In the same way, if F is a subset of edges of G, $G - F$ denotes the graph with same vertex set as V and with $E \setminus F$ as the set of edges.

A *stable set* (or independent set) of G is a subset of vertices which induces a subgraph with no edges. The *stability* of G, denoted by $\alpha(G)$, is the cardinality of a maximal stable set of G. A *clique* of G is a subset of vertices which induce a subgraph with all possible edges. The *clique* of G, denoted by $\omega(G)$, is the cardinality of a maximal clique of G. We denote by K_n the complete graph on n vertices, that is the graph consisting of a clique on n vertices.

A *k-colouration* of G is a partition of its vertices into k stable sets. The smallest integer k for which G admits a k-colouration is called the *chromatic number* of G, and is denoted by $\chi(G)$.

A *path* P is a subgraph of G, with vertices $\{x_0, x_1, \dots, x_p\}$, $p \geq 0$ and edges $\{x_i x_{i+1} : i = 0, \dots, p-1\}$. We say that P is a path from x_0 to x_p and write $P = x_0 x_1 \dots x_p$, the integer p being the *length* of P. The *endpoints* of P are x_0 et x_p, x_0 being the *beginning* of P and x_p its *end*. We denote by P_n the graph consisting of a path on n vertices.

A *cycle* C of G is a subgraph of G with vertices $\{x_1, x_2, \cdots, x_l\}$, $l \geq 1$ and edges $\{x_i x_{i+1} : i = 1, \dots, l-1\} \cup \{x_l x_1\}$. The number of edges l is the *length* of C. If all the vertices of C are distinct, then C is called a *circuit* of G. A graph that does not contain any circuit is said to be *acyclic* and is called a *forest*. The *girth* of a graph is the minimum length of a circuit and is denoted by $g(G)$. A *chord* of a cycle is an edge between two of its vertices that are not consecutive in the order. A *hole* in a graph is a chordless cycle of length at least 4 . A *k-hole* is a hole of length k. A hole is *even* or *odd* according to the parity of its length. We denote by C_n the graph consisting of a ccyle on n vertices.

Connectivity A graph is said to be *connected* if for any two vertices, there exists a path in the graph from one to another. A *connected component* of a graph is a maximal connected subgraph. It is straightforward to check that the connected components of a graph form a partition of its vertices. A connected forest is called a *tree*. In a connected graph G, a subset S of nodes and/or edges is a *cutset* if its removal disconnects G. A node set $S \subseteq V(G)$ is a *clique cutset* if it is a cutset of G and it induces a clique in G.

Forbidding substructures Given a fixed graph F, a graph G is *F-free* if it does not contain F as an induced subgraph. For a family of graphs $\mathcal{F}$, we say that G is *$\mathcal{F}$-free* if it is F-free for every $F \in \mathcal{F}$. If $\mathcal{F}$ is a family of graphs, we will denote by $\mathrm{Forb}(\mathcal{F})$ the class of $\mathcal{F}$-free graphs.

A class of graphs is *hereditary* if it is closed under induced subgraph containment. That is, if G belongs to the class, any subgraph of G also belongs to the class. Equivalently, for every G belonging to the class and for every vertex v of G, the graph $G \setminus v$ belongs to the class.

For every hereditary class $\mathcal{C}$, there exists a unique minimal family $\mathcal{F}$ of graphs for which $\mathcal{C} = \mathrm{Forb}(\mathcal{F})$. $\mathcal{F}$ is the list of minimal (with respect to the induced subgraph partial order) graphs that are not in $\mathcal{C}$. Note that this list might be infinite (the induced subgraph relation is not a well quasi order).

A weaker condition than being $\mathcal{F}$-free is to ask that the neighbourhood of every vertex induces a $\mathcal{F}$-free graph, we call such a graph *locally $\mathcal{F}$-free*.

An even weaker one is the existence, for every subgraph of G, of a vertex whose neighbourhood is $\mathcal{F}$-free. Note that this is equivalent to the existence of a *$\mathcal{F}$-elimination ordering*, that is an ordering $(v_1, \dots, v_n)$ of the vertices of a graph G such that for every $i = 1, \dots, n$, $N_{G[\{v_1,\dots,v_i\}]}(v_i)$ is $\mathcal{F}$-free.

A vertex is *simplicial* if its neighbourhood is S_2-free (where S_2 denotes the stable set on two vertices), or equivalently induces a clique. An S_2-elimination ordering is often called a *simplicial ordering* or *perfect elimination ordering*. It is a classical theorem to prove that the graph admitting such an elimination ordering are exactly chordal graphs.

Algorithms In all complexity analysis of the algorithms, n denotes the number of vertices of the input graph, and m the number of edges. We say that an algorithm runs in *linear time* if its complexity is $O(n+m)$.

One work presented in Chapter 5 concern approximation algorithms. A *Polynomial Time Approximation Scheme* (or PTAS) is an algorithm which takes an instance of an optimisation problem and a parameter $\epsilon > 0$ and, in time $n^{f(\epsilon)}$, produces a solution that is within a factor $(1+\epsilon)$ of being optimal (or $(1-\epsilon)$ for maximisation problems). An Efficient Polynomial Time Approximation Scheme (or EPTAS) is a PTAS where the running time is required to be $f(\epsilon).n^c$, where the constant c is independent of ϵ. For every such denomination, one can also define randomized version, where the algorithm outputs the solution with high probability.

Another section in Chapter 5 concerns *fixed parameter tractable* algorithms (in short FPT algorithms). This theme has been a huge area of research in the past decade, let us rapidly define the basics, see [DF13] for a book of reference on the subject. For algorithmic problems whose complexity is not known to be polynomial in the size of the input, one can try find polynomial algorithms for restricted inputs, and parametrized complexity theory aims at classifying the inputs using some parameter k (for example, the treewidth of the graph, or size of the optimal solution). A first progress is to have for each value of k, an algorithm that is polynomial in the size of the input (class XP). An algorithm in time $O\big(n^k\big)$ would be of this kind if k is the parameter and n the size of the instance (think of maximum independent set where k is the size of the optimal solution, one can try all sets of size k). An FPT algorithm is something much better : it corresponds to an algorithm in time $O\big(f(k)n^{O(1)}\big)$, where f is any function (typically a huge exponential). In between FPT and XP, there is a hierarchy $W[t]$ (which we will not define here) of complexity classes for parametrized problems for which $FPT = W[0]$. Similarly to the hypothesis $P \neq NP$ it is usually assumed that FPT is not equal to $W[1]$ and some problems, like deciding the existence of a k-clique, are shown to be W[1]-complete.

Chapter 2

Some Hereditary Classes appearing in this Document

2.1 Classes coming from the world surrounding perfect graphs

Amongst the hereditary properties that have been studied over the last decades, the one that has drawn the most attention is probably the class of perfect graphs. One could write a whole book just about this topic so of course within a page we will just expose very basic facts about them. See for example [Tro13] for an excellent survey.

A graph G is *perfect* if for every induced subgraph H of G, the chromatic number of H is equal to the maximum size of a clique of H (note that the maximum size of a clique is obviously a lower bound on the chromatic number). In 1961 [Ber61], Berge conjectured that the complement of a perfect graph is perfect. This was known as the Perfect Graph Conjecture and Lovász gave a proof of it in 1972 [Lov72]. Moreover, it is easy to see that odd holes and their complement are not perfect, so perfect graphs cannot contain any such object as an induced subgraph. Berge then also conjectured that this is sufficient to be perfect : a graph G such that both G and $\overline{G}$ are odd-hole-free (these were called *Berge* graphs in the literature) must be perfect. This was known as the Strong Perfect Graph Conjecture and was one of the most studied questions in graph theory until a proof of it was announced in 2002 by Chudnovsky et al. The proof was based on a decomposition theorem for the class of perfect graphs (the notion of decomposition theorems will be discussed more deeply in Chapter 3, so let us be informal right now).

On the algorithmic side, it was known since the eighties that maximum clique, maximum independent set and the vertex colouring problems could all be solved for perfect graphs in polynomial time using the ellipsoid method [GLS88]. It is also known that perfect graphs can be recognised in $\mathcal{O}(n^9)$-time [Chu+05b]. One fundamental question that remains in this area is how one can design purely combinatorial algorithms for the mentioned optimisation problems for perfect graphs (so avoiding the ellipsoid method). This has been recently proved for Berge graphs not containing an induced C_4 [Chu+18].

Along the way leading to the Strong Perfect Graph Conjecture, the techniques and strategies based on decompositions that would eventually be successful were applied on similar classes. The class of even-hole free graphs - graph with no induced cycle of even length - got a first decomposition theorem which led to a polynomial time recognition algorithm [Con+02a]. For the class of odd hole free graphs (which contain perfect graphs), there was a structural decomposition theorem [CCV04a], but because of some of the decomposition operations involved, this could not be used to devise a polynomial time recognition algorithm. This question was open for a long time and people even suspected it might be NP-complete (for example Bienstock [Bie91] showed that testing if a graph has an odd hole containing a given vertex was NP-complete). However, a proof of polynomiality was announced very recently [CSS19].

The complexities of finding a maximum independent set and an optimal colouring are not known for even hole-free graphs nor for odd hole-free graphs. In Section 3.4 we will expose some results concerning this problem for subclasses of even hole free graphs. Finding a maximum clique for odd hole-free graphs is NP-complete [Pol74]. On the other hand, one can find a maximum clique of an even-hole-free graph in polynomial time, since as observed by Farber [Far89] C_4-free graphs have $O(n^2)$ maximal cliques and hence one can list them all in polynomial time. In Chapter 4 we will explain how some work we did on graph searching, combined with other results could give a $O(nm)$ algorithm for even-hole free graphs (in fact for a larger class). In Section 5.2 we also discuss maximum clique in some generalisations of C_4 graphs (in fact we work in that section in the complement graphs (graphs that exclude two independent edges) so it will be presented in this section as the problem of finding a maximum independent set).

Finally, since Berge graphs satisfy $\chi = \omega$, it is natural to wonder whether there is a relation between χ and ω for odd-hole free graphs, even hole free graphs or other hereditary classes. This is a very vast topic and we will study this in Chapter 6.

Other related classes Some classical subclasses of perfect graphs will appear in this document. As already mentionned *chordal graphs* are graphs which are C_n-free for $n \geq 4$. As they clearly contain no odd holes nor odd antihole, they are Berge and hence perfect by the strong perfect graph theorem, but their perfection was known before, it follows easily from the fact that they have always a clique cut.

Another class is the class of *comparability graphs* that are graphs that can be oriented transitively : for each edge one chooses an orientation and it must satisfy that if $x \to y$ and $y \to z$ then there must be an edge between x and z and it must be oriented towards z. Their perfection was also known before the Strong Perfect Graph Theorem. A *cocomparabilty graph* is the complement of a comparability graph.

2.2 Intersection Models

It is common to define an hereditary class through what is called an intersection model. This means the following : given a set of (typically geometrical) objects, seen as subsets of a ground set, we construct a graph whose verticex set is this set of objects, and with an edge between two vertices if the corresponding objects intersect. Intersection graphs have been studied for many different families of objects due to their practical applications and their rich structural properties [MM99; BLS99]. Note that contrary to triangle free graphs, even hole free graphs, or Berge graphs which were hereditary classes defined by the list of forbidden objects, for an hereditary class defined by an intersection model, it is already an interesting question to know a minimal or at least usable list of forbidden induced subgraphs that characterise the class.

All graphs are intersection graphs... Before describing classical geometrical intersection graphs, let us point out that formally, every graph is an intersection graph : it is enough to associate to each vertex the set of edges incident to it, and now it is clear that two such sets intersect if and only if the corresponding vertices are adjacent. This is true in fact for every edge clique cover of a graph, that is a collection of cliques of the graph that cover every edge; with one such collection at hand, the graph can be seen as the intersection graph of the sets given by, for each vertex v, the subcollection of cliques that contain v. This is not the topic of this dissertation, but there are nice questions about the minimum total size of an intersection model to represent a graph on n vertices (see [EGP66]). One that I would like to mention and advertise because I tried unsuccessfully to solve it is Conjecture 2.2.1 stated below. A claw is the graph $K_{1,3}$. The class of claw free graphs is an intensively studied graph class that is the subject of hundreds of research papers.

Conjecture 2.2.1

Every claw-free graph on n vertices admits a clique edge cover with at most n cliques.

It was proved recently in [JH] under the assumption that $\alpha \geq 3$ using a general decomposition theorem of claw free graph due to Chudnovksy and Seymour ([CS05a]). So it remains the case $\alpha = 2$. In this case, one can prove $2n$ (and $3n/2$ fractionally) without too much difficulty, but to go further seems quite difficult.

Geometrical Intersection Models Intersection graphs of geometric objects are very classical objects as they can serve as a model for many concrete applications. Computing a maximum number of disjoint elements in a collection of geometrical objects (that is computing maximum independent set for the intersection graph) is amongst the oldest problems in computational geometry and its applications vary from frequency assignment in cellular networks [CCJ90; Eve+03], map labelling in computational cartography [AKS98; VA99], interval scheduling in manufacturing [Spi98; WJ+07] and chip manufacturing [HM85].

d-Ball graphs are intersection graphs of balls in $\mathbb{R}^d$, for some fixed integer d. When all balls have the same radius we speak of *unit-d-ball graphs*. For $d = 1$ they are usually referred to as *interval graphs* and when $d = 2$, they are called *disk graphs*.

Interval Graphs are a very classical family, and have been the subject of many papers and arise naturally in real life scheduling optimization problem [MM99; BLS99]. Here are some facts about them.

- There exists a characterization by (infinitely many) forbidden subgraphs [LB62]
- They are exactly graphs that are both chordal and co-comparability.
- There are recognition algorithms in linear time (that do not use the characterization in terms of forbidden structures but the the linear order representation explained in section 2.3),
- all classical NP-Hard problems are polynomially solvable on this class.

For Disk or Ball Graphs, it is to note that although some minimal graphs not in the class are known, such as $K_{1,7}$, there is no known characterization by forbidden subgraph (neither for unit disk graphs). On the algorithm side, let us mention :

- Recognition : it is NP-Hard to decide if an input graph is a disk graph, or a unit disk graph ([HK01] [BK98]). Additionally, it is provably impossible in polynomial time to output explicit coordinates of a unit disk graph representation: there exist unit disk graphs that require exponentially many bits of precision in any such representation.
- In Chapter 5 we will review the complexity for Maximum Independent Set and Maximum Clique for these classes, so we will leave that aside for the moment
- Colouring them is also hard. Clark et al proved that 3-colouring is already NP-Hard [CCJ90]. It was extended to k-colouring for any k by Gräf et al. [GSW98].

Circle Graphs are defined as intersection graphs of chords on a circle in the plane. They have been introduced by Even and Itai in [EI71], to solve an ordering problem with the minimum number of parallel stacks without the restriction of loading before unloading is completed, proving that the problem can be translated into the problem of finding the chromatic number of a circle graphs. They have many applications, from container ship stowage [APS00] and reconstruction of long DNA strings from short subsequences [Arr+00].

The recognition problem of circle graphs was asked by Golumbic in [Gol04] an was eventually settled independently by Naji [Naj85], Bouchet [Bou94], and Gabor et al. [GHS89]. The best current running time is the algorithm of Gioan et al. [Gio+14a]

Bouchet's algorithm emphasized the relation of circle graphs with the notion of vertex minor. A graph H is a vertex minor of a graph G if it can be obtained from an induced subgraph of G by repeatedly applying the following operation : choose a vertex v and complement the graph formed by its neighbours. It is easy too see circle graphs are closed under taking vertex minor as for every circle graph and every chord AB in a geometrical model of it, by completely reversing the order of the vertices on one of two circle arcs limited by A and B, one produces a model for a graph that is exactly the graph obtained by complementing the neighbourhood of the vertex represented by the chord AB. Bouchet's algorithm is based on a proof ([Bou94]) that a graph is a circle graph if and only if it does not contain three small graphs as a vertex minor. The notion of vertex minor is strongly connected to that of rankwidth, [Oum17]), where circle graphs play role similar to that of planar graph with respect to tree-width. It is conjectured, that every graph with sufficiently large rankwidth should contain a vertex minor of any fixed bipartite circle graph. All algorithms take advantage of some properties of the circle graphs with respect to their split decomposition, which we will discuss in Chapter 3.

Other Classes Characterized by Intersection Models Several hereditary classes not defined directly through an intersection model do have a characterization theorem in such terms. Let us mention some classical ones. Recall that chordal graphs are graphs that contain no induced cycle of length at least 4; a *permutation graph* is a graph associated to a given permutation on n elements. Its vertices represent the elements of a permutation, and two vertices are liked by an edge if the corresponding elements are reversed by the permutation. Some of the results below are absolutely non trivial. The fourth one is the celebrated Circle Packing Theorem (or Koebe-Thurston-Andreev [Koe36; And70]).

The fifth one was formerly known as Scheinermann Conjecture before its resolution by Chalopin and Goncalves in 2009 ([CGO10]). The first one comes from the fact that chordal graphs have clique cutsets (and the Helly property of subtrees of a tree).

- A graph is chordal if and only if it is an intersection graph of subtrees of a tree.
- A graph is permutation if and only if it is intersection graph of line segments whose endpoints lie on two parallel lines
- A graph is a co-comparability if and only if it is the intersection graph of curves from a line to a parallel line
- A graph is planar if and only if it is the intersection graph of a collection of disks in the plane whose interior a pairwise disjoint (they form tangent circles).
- If a graph is planar then it is the intersection graph of line segments in the plane.

2.3 Forbidding Linear patterns

We have already defined in 1.4 the notion of F-elimination ordering : an ordering of the vertices such that for every vertex the set of its neighbours that precede it in the order do not induce a copy of F (a simplicial ordering being an $\overline{K_2}$-elimination ordering). This is in fact a particular case of a nice way to define hereditary class that, although will not be effectively studied in this document, I will take a page to discuss as I believe it is an interesting direction of research. An *ordered graph* is a graph given with a total order on its vertices. Given a family $\mathcal{F}$ of ordered graphs, the hereditary class $\mathrm{LinForb}(\mathcal{F})$ is defined as the set of graphs for which there exists an ordering of the nodes, such that no induced ordered subgraph is isomorphic to an ordered graph in $\mathcal{F}$. If $\mathcal{F}$ consists of only one ordered graph G we will just write $\mathrm{LinForb}(G)$ instead of $\mathrm{LinForb}(\{G\})$.

If the last vertex of every ordered graph in $\mathcal{F}$ is universal for this graph, then this notions corresponds to elimination orders. For example, the result stating that chordal graphs are characterized by the existence of a perfect elimination order says exactly that the class of chordal graphs is $\mathrm{LinForb}(P)$, where P is the induced path on three vertices ordered in such a way that the middle vertex is last. If the 3 vertices are ordered such that the middle vertex is in the middle then forbidding it yields the class of comparability graphs.

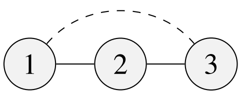
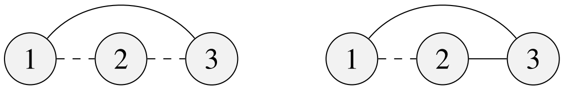


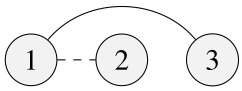
Figure 2.1: a forbidden ordered graph for comparability graphs

Let us consider another classical hereditary class of graphs : interval graphs, which are defined as intersection graphs of intervals on a line. There exists (see [LB62]) a characterization in terms of forbidden induced subgraphs, and it is quite complex, as it relies on forbidding three graphs and three infinite families. In particular, this characterization does not give a good recognition algorithm. On the other hand it has the following characterization by orderings : $G = (V, E)$ is in an interval graph if and only if there exists an ordering of the vertices, denoted by $<$, such that there is no triple of vertices (u, v, w), with $u < v < w$, $uw \in E$ and $uv \notin E$ [RR88; Ola91]. With this vocabulary, the class of interval graphs is the class defined by forbidding the two ordered graphs below

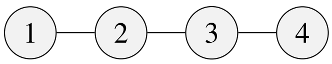


This characterization of interval graphs is not only simple to express but also useful since there exists good recognition algorithm that take advantage of it ([COS09]).

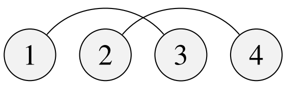
In the example for interval graphs above, we see that the presence of an edge between the two vertices 2 and 3 is irrelevant, so sometimes to simplify things we simply say that we forbid an ordered trigraph, that is a graph with edges, non edges, and undecided edges, meaning that we forbid all ordered subgraphs that are realizations of this trigraphs (a realization of a trigraph is any graph where the undecided edge have been given a status : either edge or non edge). So interval graphs are $\mathrm{LinForb}(Q)$, where Q is the following pattern :



There are many other interesting examples and this is not necessarily well known so let us present two more. If P_k is the ordered trigraph on $k + 1$ vertices such that $x_i x_{i+1}$ is an edge and every other edge is undecided, then $\mathrm{LinForb}(P_k)$ is the class of graphs of chromatic number at most k (this can be easily generalized for (a, b)-colourings : partitions into a stable sets and b cliques). Forbidding the pattern below corresponds to 3-colourable graphs.



One can also prove that outerplanar graphs are exactly $\mathrm{LinForb}(P)$ where P is the pattern below.



This approach of looking at hereditary classes through vertex ordering has been investigated in [Woo04]. It has been recently applied to characterize graphs with bounded asteroidal number [CS15] and seems to be promising. Another interesting application is to prove that a given graph class is closed under some operations, as for example in [BH08], to prove that the square of the line-graph of a chordal graph is chordal, or in [HM17a] to design algorithms for maximal induced matchings. We believe certain hereditary graph classes that have difficult characterizations in terms of forbidden induced subgraphs (like for instance path graphs, see [LMP09]) might have simpler ones using these vertex orderings.

Recognition problem are also interesting. As shown by the example of 3 colourability being defined by linearly forbidding a 3 edge path, we see that deciding some small patterns can be NP-Hard. It was shown in [HMR14] that for families of patterns on 3 nodes, everything is polynomial (see also [FH18]). On the other hand, it was shown in [DGR95] that almost all classes defined by forbidding 2-connected patterns are NP-complete to recognize. In [HMR14], the authors conjecture a dichotomy on this type of recognition problem. Recent results on this topic can be found in [FH18]

Chapter 3

Decomposing

3.1 Introduction

One strategy that has revealed to be of great strength for both theoretical purposes and for design of algorithms for hereditary classes of graphs is the theory of graph decompositions. Proving a decomposition theorem for a hereditary class $\mathcal{C}$ consists in describing a subset $\mathcal{C}_0$ (often called "basic class") of $\mathcal{C}$ and a certain list $\mathcal{L}$ of graph compositions (example of which we will see just after), such that if $G \in \mathcal{C}$ then

- either G belongs to $\mathcal{C}_0$,
- or G can be built from smaller graphs G' and G'' belonging to $\mathcal{C}$ using an operation in $\mathcal{L}$.

Such a theorem says that the graph can be decomposed in a tree-like fashion, internal nodes corresponding to decompositions in $\mathcal{L}$ and leaves to graphs in $\mathcal{C}_0$.

Now suppose that one wants to prove the inclusion of a class $\mathcal{C}$ into another class $\mathcal{C}'$ (which is in a way what most theorems are saying) then with such a theorem for class $\mathcal{C}$ in hand, it is enough to prove :

- that basic graphs are in $\mathcal{C}'$
- that the prescribed compositions operations preserve the fact of being in $\mathcal{C}'$.

That approach has proved very successful in the last decades, one celebrated example being the proof of the long standing Strong Perfect Graph Conjecture by Chudnovsky, Robertson, Seymour and Thomas ([Chu+06]), which consists in the (difficult) proof of such a structure theorem for the class of Berge Graphs. Deep decomposition theorems exist also for the class of even-hole free graphs, odd-hold free graphs, claw-free graphs, to name a few.

It should also not come as a surprise also that in order to design algorithms such a decomposition theorem will be of use, in a kind of "divide and conquer" approach. This is is useful both for efficiently solving a decision problem for graph in the class (does my input graph in the class has chromatic number at most k) but also to design a recognition algorithm for the class (if the input graph is not basic and does not contain any of the decomposition, then it is not in the class). It is also the case that the same decompositions, that we will discuss more deeply later in the document, appear in many decomposition theorems. It is therefore an important issue to have the best possible algorithms to find those particular decomposition, if they exist.

Let us illustrate this paradigm with one of the easiest such theorems : the one characterizing *cographs* (or complement-reducible graph). These have been discovered and described independently by various authors in the 1970s, see [CLB81]. They can be defined as P_4-free graphs, i.e. graphs that do not contain the path P_4 as an induced subgraph.

Theorem 3.1.1

If G is a P_4-free with at least two vertices, then either G or its complement is disconnected.

This is exactly the kind of theorem mentioned above : if G is P_4-free, then either G is K_1, or G can be obtained from two graphs G' and G'' by either a disjoint union, or a complete join (i.e. obtained by the disjoint union by adding all edges between G' and G'').

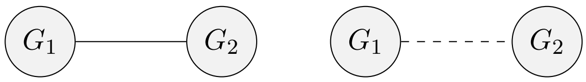


Figure 3.1: A complete join and a disjoint union

If G is a cograph then it can be therefore recursively decomposed using these two operations and to a cograph hence corresponds a rooted binary tree, whose leaves are labelled bijectively by the vertices of G and where each internal node is labelled by either 1 (meaning complete cut) or 0 (disjoint union). Two vertices are adjacent if and only if the least common ancestor of the corresponding leaves is labelled 1.

Note that, one can merge internal nodes that are adjacent and have the same label (the tree is not binary anymore) so that it still encode the graph with this least common ancestor rule. Each time one block of decomposition is not connected (resp. has a disconnected complement) one just put one children per connected component (resp. connected component of its complement).

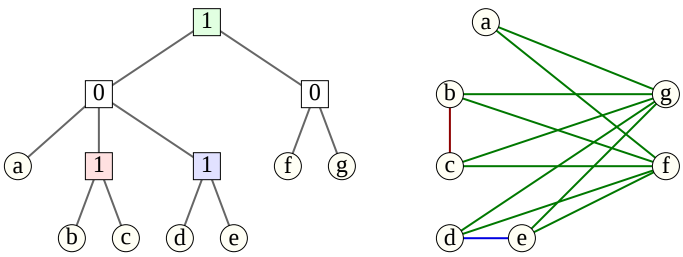


Figure 3.2: A cotree and the corresponding cograph

To every internal node t of this tree one can associate the set of vertices X_t of G that correspond to leaves that are below t. It is easy to see that X_t is such that every vertex not in X_t is either adjacent to every vertex of X_t or to none. Such a set in a graph is called a module. In the cograph case, it is also not difficult to to see that the cotree encodes all the modules of the graph, in the sense that every module is either some X_t or the union of some X_{t_i} where the t_i are children of the same node.

Note that this scheme of decomposition (either G or its complement is disconnected) can be applied to any graph, even it is not a cograph, the difference is that some internal nodes cannot be decomposed anymore. But is it possible to push this further and fully decompose the graph by adding new kind of decompositions apart from connected components or connected components of the complement? Can one get also a compact tree representation of all modules of a graph? It turns out that one can do both at the same time, and also that the decomposition tree can be computed in linear time. We will discuss this shortly in the next section before exploring other decompositions.

General Edge Cut Decompositions Note that the decomposition operations involved above - disjoint cut, compete cut, modules - are of the edge cut type : the vertices of G are partitioned into two sets, and the shape of the edge cut is constrained. More precisely if M is a $0, 1$ matrix with p rows and q columns we say that a partition of the vertices (A, B) is of type M if there exists a partition of A in $A_1, A_2, \ldots, A_p$ and B in $B_1, B_2, \ldots B_q$ such that if $(a, b) \in A_i \times B_j$, then ab is an edge if and only if $M_{ij} = 1$. The case of disjoint union or complete join involved in the cograph theorem corresponds to a matrix with 1 row and column and entry equal respectively to 0 or 1. The case of module is the case of a matrix with one row and two columns with one 0 entry and one 1 entry.

Following this scheme, one next case after modules corresponds to a matrix on two rows and columns with only one non zero entry. This is exactly what is called a *split*, or 1-join. As we will explain in section 3.3, to this decomposition corresponds a nice structural theorem for which we have obtain in [CDR12] a fast decomposition algorithm.

The fourth section in this chapter will be devoted to a work on 2-join, which is the case of a 3x3 matrix where only 2 entries are equal to 1 (not on the same line or column, or else it would just be a split). It is another kind of graph decomposition that appeared in several decomposition theorems, such as the decomposition of Berge graphs in the proof of the Strong Perfect Graph Theorem, or the decomposition theorems for even hole free graphs. In [Cha+12], with Habib, Trotignon, and Vušković, we improved all previously known algorithms to detect various kinds of 2-joins, we will discuss in this section the ideas of these results and their applications. In [HMD14], Mamcarz et al. pushed the ideas of [Cha+12] to investigate decomposition theorems in the case of general matrices M.

3.2 Modular Decomposition

Recall that X is a *module* if for every $x \notin X$, either $N(x) \cap X = \emptyset$ or $N(x) \supset X$. Modular decomposition, was introduced by Gallai ([Gal68]) in his work on transitive orientations of comparability graphs. As we will rapidly explain (this is not the topic of this section, see [HP10] or [Pau06] for an very good survey on the notion), to any graph corresponds a labelled tree with at most $|E(G)|$ nodes that describes the structure of the whole family of modules in a graph G. This is a well-studied structure and furthermore there exists algorithms (see [CH94; MS94b] in time $O(n + m)$, to compute this decomposition tree. Historically, the notion reappeared in the literature under other names (a module is also called an *homogeneous set*) and notably in several decomposition theorems, especially in the theory around perfect graphs ([Chu+06]) and claw-free graphs ([CS08b]). This decomposition is also used in the design of efficient algorithms (see for instance [MR84; BLS99]).

Of course there can be exponentially many modules in a graph (in a complete graph, every subset of vertices is a module), but there is in fact a compact representation (meaning linear in the size of the graph) that represents all modules and their inclusions. The crucial property behind this fact is modules form a so called *partitive* family. Before giving the definition, let us from now on say that two sets *overlap* if they intersect and none is included in the other.

Definition 3.2.1

Let $\mathcal{F}$ be a family of subsets of a ground set V. $\mathcal{F}$ is partitive if

- *V and all singletons belong to $\mathcal{F}$*
- *for all $X, Y \in \mathcal{F}$ that overlap, $X \cup Y$, $X \cap Y$, and $(X \setminus Y) \cup (Y \setminus X)$ are also in $\mathcal{F}$*

In a partitive family it is useful to look at the so called *strong* elements in the family : the ones that do not overlap any other member of the family. This subfamily is *laminar* (i.e. no two sets overlap) and this is straightforward to see that any laminar family can be represented by a rooted tree whose leaves are in bijection with the elements of the ground set and such that every member of the family corresponds to the set of leaves that are descendant of an internal node.

Now it is not difficult to see that every partitive family is fully represented by the tree of its strong elements by adding some labels on the internal nodes. First any element of the family must consist of the union of some strong elements that are represented by children of the same node (otherwise it would overlap some strong element). Moreover for a strong node it is not difficult to prove that either every union of at least 2 of its children is in the family or no such union is in the family. This leads to the following definition : a *partitive tree* is a rooted tree T whose internal nodes are labelled *Prime* or *Complete*, and whose leaves are labelled in bijection with the elements of V. We associate with such a tree the family of subsets of V, that are of three kinds.

- For every Prime node of the tree : the subset of V consisting of all (vertices represented by) leaves that are descendants of this node,
- for every Complete node, and for every possible union of its children : the union of the subsets of V represented by these children,
- for every leaf of the tree : the corresponding singleton.

The previous discussion gives the proof of the following theorem.

Theorem 3.2.2 (*[CHM81; Möh85; HM03]*)

Any partitive family can be represented by a partitive tree.

In the case of modules of a graph, the complete nodes corresponds to strong modules that induce a graph that is either disconnected, or whose complement is disconnected (they are usually labelled respectively series or parallel nodes). The Prime nodes correspond to strong modules whose quotient graph (the graph obtained by replacing every children module by a single vertex) is a prime graph,

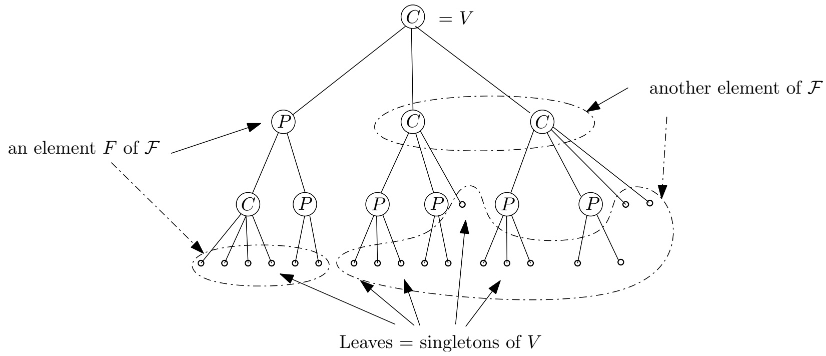


Figure 3.3: Partitive family

that is a graph that does not admit any non trivial module - or equivalently graphs that are both connected and co-connected. As we said before, the important fact is that this tree, called modular decomposition tree, can be computed in linear time ([CH94; MS94b]).

3.3 Split Decomposition

3.3.1 Preamble

The *split* decomposition, also known as 1-*join* decomposition, is a generalization of modular decomposition, that has a large range of applications, from NP-hard optimization [Rao04; Rao08] to the recognition of certain classes of graphs such as distance hereditary graphs [GP03; GP07], circle graphs [Spi94] and parity graphs [CS99; Dah00b]. A survey of applications of the split decomposition in graph theory can be found in [Rao08]. This decomposition was introduced by Cunningham in [Cun82] who also presented the first worst case $O(n^3)$-time algorithm. The complexity was improved to $O(nm)$ in [GHS89] and to $O(n^2)$ in [MS94a] (n being the number of vertices and m the number of edges of the graph). One should also mention the quasi linear algorithm by Gioan et al. ([Gio+14b]) that also gives a recognition algorithm for the class of circle graphs (intersection of chords of a circle).

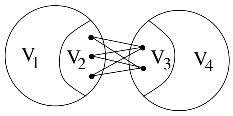
Two papers have been written by E. Dahlhaus on solving the problem in linear time: an extended abstract in 1994 [Dah94] followed several years later (in 2000) by an article in Journal of Algorithms [Dah00b]. However, while these two manuscripts substantially differ, they are both very difficult to read, and the algorithm presented is so involved that its proof and linear-time complexity are quite difficult to check. This motivated de Montgolfier, Raffinot and myself to work on the question and in [CMR09] we gave a new $O(m+n)$-time algorithm to solve this problem. We develop in the paper some theoretical ideas and methods that result in a much more comprehensive and well-founded algorithm. The next subsections will be devoted to the description of these ideas.

3.3.2 Splits

Definition 3.3.1

A $split$ of $G=(V,E)$ is a partition of V into two non-empty subsets X_1 and X_2 such that the edges between X_1 and X_2 induce a complete bipartite graph.

In other words, there exists a partition of V into 4 subsets V_1, V_2, V_3, V_4, such that $X_1 = V_1 \cup V_2$ and $X_2 = V_3 \cup V_4$, and such that G contains all possible edges between V_2 and V_3, and no other edges between X_1 and X_2.
We denote splits either by bipartitions (X_1, X_2) or by quadripartitions (V_1, V_2, V_3, V_4) depending on needs. Both are equivalent since there is a unique quadripartition for each bipartition.
A split is said to be non trivial *if both sides have more than two vertices.*



Structure of a split.

Figure 3.4 illustrates the notion of a split. We see that a module is a special case of split when V_1 is empty.

A graph may contain an exponential number of modules (and therefore splits) but, as in the case of modules, all splits may be represented in a compact way. Again the key is in the structure of its "strong elements", whose definition we give now. Two splits (V_1, V_2, V_3, V_4) and (V_1', V_2', V_3', V_4') *cross* if $V_1 \cup V_2$ overlaps both $V_1' \cup V_2'$ and $V_3' \cup V_4'$. A split is *strong* if it crosses no other split.

The following theorem of Cunningham (see [Cun82]) tells us that there exists a labelled tree that encodes all splits of a graph.

Theorem 3.3.2 (*Cunningham, 1982 [Cun82]*)

For any graph G, there exists a labelled tree T called Split Tree of G such that

- *Every leaf is labelled by a vertex of G, and this labelling constitutes a bijection.*
- *The edges of T are in bijection with the strong splits of G : to every edge e of T is associated the bipartition of $V(G)$ given by the leaves belonging to each of the two components of $T \setminus e$.*
- *Every internal node is labelled either Prime, Clique or Star, and for Star Nodes one of the incident edges is out oriented to a neighbour called the center of the star.*
- *If the node t of T has degree k and $V_1, V_2, \ldots V_k$ is the partition of $V(G)$ induced by the k incident edges, then*
  - *If t is labelled Complete, for any subset $I \subset \{1, \ldots, k\}$, $(\cup_{i \in I} V_i, \cup_{i \notin I} V_i)$ is a split and there is an edge between all pairs of distinct V_i*
  - *If t is labelled Star for any subset $I \subset \{1, \ldots, k\}$, $(\cup_{i \in I} V_i, \cup_{i \notin I} V_i)$ is a split. Furthermore, if (w.l.o.g) V_1 is the center, then there is an edge between V_i and V_j if and only if $i = 1$ or $j = 1$.*
- *Every split of G is described by one of the cases described above.*

The names Prime, Complete, and Star come for the three possibles shapes the quotient graph (the graph obtained for the partition $(V_1, V_2, \ldots, V_k)$ by replacing each part with a vertex v_i with an edge between v_i and v_j if and only if there is one between V_i and V_j. Either this graph is prime (it has no non trivial splits) or it is a clique or a star.

The Split Tree defined in this theorem is the object we are aiming for and that will be constructed by our algorithm.

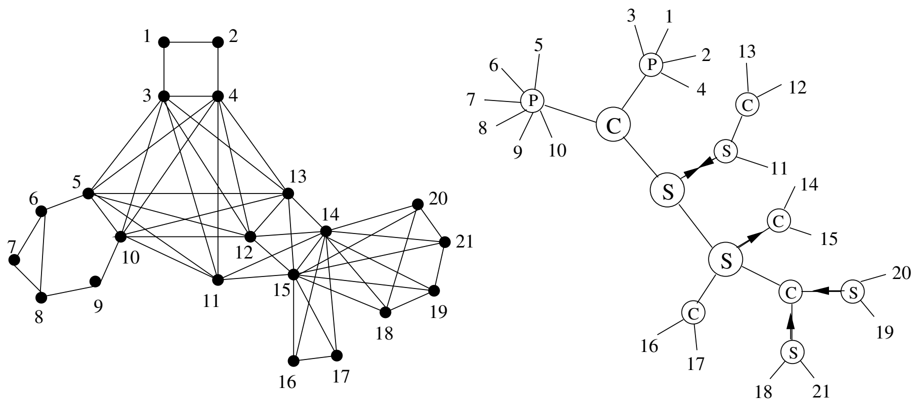


Figure 3.5: An example of graph and its corresponding *split tree*. Nodes labelled C, S and P are respectively clique, star and prime. An orientation is associated to each star node to point its center. Note that nodes with 3 incident edges could have been labelled *prime*

.

In order to design our algorithm, we will now break the symmetry in the definition of splits by picking arbitrarily a root $r \in V$, and defining a family of subsets consisting of the side that does not contain r, and this for two reasons :

- By doing this, we get a partitive family and we will strongly exploit this fact (and this in fact proves Cunningham theorem above).

- The way our algorithm works is precisely to decompose the graph into layer distances from the root (using a breadth-first search algorithm) and then compute the things we are aiming for in a layer by layer bottom-up approach.

From now on, we assume the root r is fixed, and for any given split (V_1, V_2, V_3, V_4) (using notations of Definition 3.3.1) we assume without loss of generality that we always have $r \in (V_1 \cup V_2)$. The set $V_3 \cup V_4$ is called the *split bottom* and the set V_3 is called the *split border* of the split (V_1, V_2, V_3, V_4). Notice that two different splits bottoms may share the same border.

The easy to check but crucial fact behind Theorem 3.3.2 is that split bottoms form a partitive family. And if we have in hand the partitive tree representing split bottoms, the Split Tree described at Theorem 3.3.2 is then obtained by adding a leaf labelled by r at the root of the tree. Regarding labels, nodes labelled Prime will be keep their label and the Complete nodes will either correspond to Clique or Star nodes.

3.3.3 A Tool : The Orthogonal Family

We present now another way of seeing partitive families that will be central for our algorithm. When two subsets X and Y do not overlap, we will say that they are *orthogonal*, and denote this by $X \perp Y$. Now if $\mathcal{F} \subset 2^V$ is a family of subsets of V we define its *orthogonal family*, denoted by $\mathcal{F}^\perp$, by

$$\mathcal{F}^\perp = \{X \subseteq V \mid \forall Y \in \mathcal{F},\ X \perp Y\}.$$

The important properties of the orthogonal are summarized in the Proposition below:

Proposition 3.3.3

Let $\mathcal{F}$ be a family of subsets of a ground set V.

1. $(\mathcal{F} \cup \mathcal{F}')^\perp = \mathcal{F}^\perp \cap \mathcal{F}'^\perp$
2. *$\mathcal{F}^\perp$ is a partitive family.*
3. *If $\mathcal{F}$ is partitive, then the tree representation of $\mathcal{F}^\perp$ is obtained from that of $\mathcal{F}$ by switching* Prime *and* Complete *nodes.*
4. *If $\mathcal{F}$ is partitive, then $\mathcal{F}^{\perp\perp} = \mathcal{F}$. Therefore, every partitive family $\mathcal{F}$ is the orthogonal of some family $\mathcal{F}'$*

The following theorem due to McConnell (in a paper about matrices with the consecutive ones properties) states that it is possible to compute the tree representation of its orthogonal in an efficient way. We use the notation $\|\mathcal{F}\| = \sum_{F \in \mathcal{F}} |F|$.

Theorem 3.3.4 (*[McC04]*)

Given a family of subsets $\mathcal{F}$, it is possible in $O(\|\mathcal{F}\|)$ time to compute the partitive tree representation of $\mathcal{F}^\perp$.

It should be noticed that this algorithm is mainly based on an algorithm of Dahlhaus for computing overlap classes, presented in [Dah00a]. We revisited, and simplified and implemented this last algorithm in [Cha+08]. The main computational insight is that although the overlap graph of $\mathcal{F}$ can be of quadratic size, the overlap components can be computed in $O(\|\mathcal{F}\|)$ time.

The main idea of our algorithm for split decomposition will be to express the family of split borders as orthogonal of some families we are able to compute and use the previous theorem.

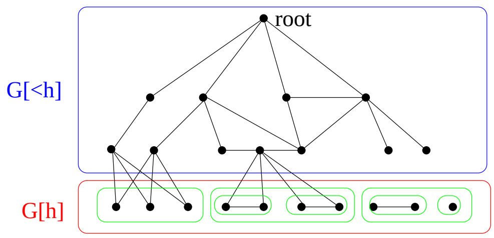
3.3.4 Structure of Split Borders

We define the *distance* of a split bottom (resp. border) S as its distance from the root, that is $\min_{x \in S} d(r, x)$. We denote $G[h]$ as the subgraph induced by the vertices at distance h, and $G[\leq h]$ as the subgraph induced by the vertices at a distance of h at most, and similarly $G[< h]$ or $G[> h]$ in the obvious way. For $X \subset G[> h]$ we denote $N_h(X)$ as the set $N(X) \cap G[h]$. Moreover, the letter H always denotes the set of vertices of $G[h]$. Note also that all orthogonal notations here refer to the orthogonal with respect to the ground set H.

An easy but important remark is the following

Lemma 3.3.5

All vertices of a given border are at the same distance from the root r.



Split Borders

This justifies the approach of our algorithm: we first compute (using a breadth first search for example) the distance layers of our graph, and then we process one layer after the other in a bottom-up approach from the furthest layer to the first one. At each step we need to determine the set $\mathcal{B}_h$ of all split borders at distance h from the root r. Let $C_1, \ldots, C_k$ be the connected components of $G[> h]$. We define two families of subsets of H:

- $\mathcal{M} = \{\text{modules of } G[\leq h] \text{ that are subsets of } H\}$,
- $\mathcal{V} = \bigcup_{i=1}^{k} \mathcal{V}_i$, where

$$\mathcal{V}_i = \{N(C_i) \cap H\} \cup \{N(x) \cap H \mid x \in C_i\} \cup \{(N(C_i) \setminus N(x)) \cap H \mid x \in C_i\}$$

The result on which the algorithm is based is then the following. The proof can be found in [CDR12].

Theorem 3.3.6

$$\mathcal{B}_h = \mathcal{M} \cap \mathcal{V}^{\perp}$$

Using Proposition 3.3.3, we can rewrite this as $\mathcal{B}_h \cup \{H\} = (\mathcal{M}^{\perp} \cup \mathcal{V})^{\perp}$ and use Theorem 3.3.4 to compute $\mathcal{B}_h$. The only thing we have to be careful is that the family $\mathcal{M}^{\perp} \cup \mathcal{V}$ might be too big but we described in our paper two tricks to circumvent this issues : we can compute efficiently smaller families $\mathcal{N}$ and $\mathcal{W}$ such that $\mathcal{N} = \mathcal{M}^{\perp}$ and $\mathcal{W}^{\perp} = \mathcal{V}^{\perp}$.

3.3.5 Sketch of the Algorithm

We explained before the structure we aim for, and we described also the main theorem behind the algorithm to obtain the split borders at a given distance. We will omit the rest of the proof, there are still many technicalities but the general idea is the following : we recursively compute a forest $\mathcal{F}_h$ that roughly represents all split bottoms at distance at least h. To compute $\mathcal{F}_h$ from $\mathcal{F}_{h+1}$, we first compute the forest representing $\mathcal{B}_h$ using the discussion above, and use this information to add new leaves to $\mathcal{F}_{h+1}$ (representing the vertices of $G[h]$), add new root nodes, and eventually merging some forests. Finally $\mathcal{F}_1$ is the tree representing of all split bottoms.This would be difficult to describe more without doing a full proof, we refer the interested reader to Sections 4 and 5 of [CDR12].

3.3.6 Continuation - Circle Graphs

Circle graphs - intersection graphs of chords on a cycle - is a widely studied class of graphs whose structure is strongly related to splits. Indeed, if a graph G admits a split with bipartition (X_1, X_2) and we call G_1 the graph obtained by contracting X_2 to a single vertex attached to the border in X_1 (and symmetrically G_2, then it is not difficult to see that G is a circle graph if and only if G_1 and G_2 are circle graphs. It can be shown as a consequence that a circle graph is prime for split decomposition if and only if it has a unique representation as a circle graph. This strong connection explains why split decomposition was key to recognition algorithm for split graphs, the first one in time $O(n^2)$ by Spinrad, and then the one in time $O(m+n)\,\alpha(n+m)$ (α denoting the inverse Ackermann function) by Gioan et al.

It is a project that we would like to pursue, there are probably chances that our algorithm and the understanding we have of the structure of splits, could help devise a $O(m+n)$ time algorithm for circle graph recognition.

3.4 A Fast Algorithm for 2-join and Some Consequences

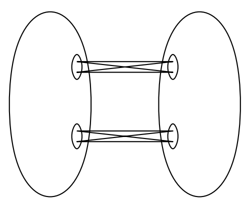
In this section we present and expose without proofs the results of [Cha+12] about 2-joins.

Definition 3.4.1

A partition (X_1, X_2) of the vertex-set of a graph G is a 2-join *if for $i = 1, 2$, there exist disjoint non-empty $A_i, B_i \subseteq X_i$ satisfying the following:*

- *every vertex of A_1 is adjacent to every vertex of A_2, every vertex of B_1 is adjacent to every vertex of B_2, and there are no other edges between X_1 and X_2;*
- *for $i = 1, 2$, $|X_i| \geq 3$.*

Sets X_1 and X_2 are the two sides *of the 2-join. For $i = 1, 2$, we will denote by C_i the set $X_i \setminus (A_i \cup B_i)$.*



A 2-join

The 2-join was first introduced by Cornuéjols and Cunningham in [CC85] in the context of studying composition operations that preserve perfection. 2-Joins ended up playing a key role in structural characterizations of several complex classes of graphs closed under taking induced subgraphs, and construction of polynomial time recognition and optimization algorithms associated with these classes. 2-Joins are used in decomposition theorems for balanced bipartite graphs that correspond to balanced $0, 1$ matrices [CCR99] as well as balanced $0, \pm 1$ matrices [Con+01a], even-hole-free graphs [Con+02a], odd-hole-free graphs [CCV04a], square-free Berge graphs [CCV04b], Berge graphs in general [Chu+06; Chu06; Tro08] and claw-free graphs [CS05b]. The decomposition theorem in [Chu+06] famously proved the Strong Perfect Graph Conjecture.

Decomposition based polynomial time recognition algorithms, that use 2-joins, are constructed for balanced $0, \pm 1$ matrices [Con+01b], even-hole-free graphs [Con+02b; DV13] and Berge graphs with no balanced skew partition [Tro08]. 2-Joins are also used in [TV12] for solving the following combinatorial optimization problems in polynomial time: finding a maximum weighted clique, a maximum weighted stable set and an optimal colouring for Berge graphs with no balanced skew partition and no homogeneous pairs, and finding a maximum weighted stable set for even-hole-free graphs with no star cutset.

In [CC85] an $O(n^3m)$ algorithm for finding a 2-join in a graph G (or detecting that the graph does not have one) is given. The algorithm is based on a set of forcing rules that for a given pair of edges a_1a_2 and b_1b_2 decides, in time $O(n^2)$, whether there exists, a 2-join with split $(X_1, X_2, A_1, B_1, A_2, B_2)$ such that for $i = 1, 2$, $a_i \in A_i$ and $b_i \in B_i$, and finds it if it does. In Section 2 of [Cha+12], we describe a new method to achieve the same goal slightly faster, in time $O(n + m)$.

Based on the fact that for any spanning tree T of G, any 2-join (X_1, X_2) must contain an edge of T that is between X_1 and X_2, it is observed in [CC85] that to find a 2-join in a graph, one needs to check $O(nm)$ pairs of edges a_1a_2 and b_1b_2, giving the total running time of $O(n^3m)$ for finding a 2-join. In Section 3 of our paper [Cha+12], we showed that actually one only needs to check $O(n^2)$ pairs of edges, reducing the running time of finding a 2-join to $O(n^2m)$.

All the 2-joins whose detection is needed for the algorithms mentioned above in fact have an additional crucial property: they are *non-path* 2-joins. A 2-join is said to be a *path 2-join* if it has a split $(X_1, X_2, A_1, B_1, A_2, B_2)$ such that for some $i \in \{1, 2\}$, $G[X_i]$ is a path with an end in A_i, an end in B_i and interior in C_i. In this case X_i is said to be a *path-side* of this 2-join. A *non-path 2-join* is a 2-join that is not a path 2-join. In [Con+02a] it is observed that by applying the 2-join detection algorithm $O(n)$ times one can find a non-path 2-join if there is one. In Section 4 of [Cha+12], we showed that in fact a constant number of calls to the algorithm for 2-join is needed, so that non-path 2-joins can also be detected in $O(n^2m)$-time.

In inductive arguments or algorithms that use cutsets, i.e. decomposition theorems, one needs the concept of the *blocks of decomposition*, by which a graph is decomposed into “simpler” graphs. Blocks of decomposition of a graph G with respect to a 2-join with split $(X_1, X_2, A_1, B_1, A_2, B_2)$ are graphs G_1 and G_2 usually constructed as follows: G_1 is obtained from G by replacing X_2 by a *marker path* P_2 that is a chordless path from a vertex a_2 complete to A_1 to a vertex b_2 complete to B_1, and whose interior vertices are all of degree two in G_1. Block G_2 is obtained similarly by replacing X_1 by a marker path P_1. In all of the above mentioned papers, blocks of decomposition for 2-joins are constructed this way, where marker paths are of some fixed small length. For example in [CC85] they are of length 1, and in the other papers they are of length at most 6. This explains why non-path 2-joins are often the useful concept and why algorithms to find non-path 2-join are needed.

In [CC85] it is claimed that at most n applications of the 2-join detection algorithm are needed to decompose a graph into irreducible factors, i.e. graphs that have no 2-join. This is true, as shown in [Con+02b], but in [CC85] it is based on a wrong observation that the 2-join detection algorithm given in [CC85] always finds an extreme 2-join, i.e. one whose both blocks of decomposition are irreducible. First of all it is not true that every graph that has a 2-join, has an extreme 2-join. For example graph G in Figure 3.8 has exactly two 2-joins, one is represented with bold lines, and the other is equivalent to it. Both of the blocks of decomposition are isomorphic to graph H (where dotted lines represent paths of arbitrary length, possibly of length 0), and H has a 2-join whose edges are represented with bold lines. So G does not have an extreme 2-join. Even if a graph had an extreme 2-join the algorithm in [CC85] would not necessarily find it.

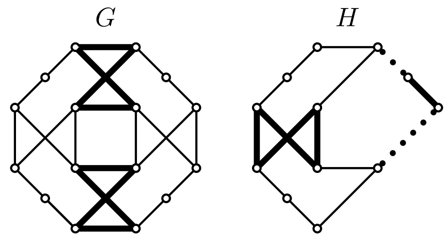


Figure 3.8: A graph G with no extreme 2-join

For the optimization algorithms in [TV12], it is in fact essential that these extreme non-path 2-joins are used, which is potentially a problem since as shown above, a graph with a 2-join may fail to have an extreme 2-join. Fortunately, graphs studied in [TV12] have no star cutset, where a *star cutset* is any set $S \subseteq V(G)$ such that $G \setminus S$ is disconnected and for some $x \in S$, x is adjacent to all vertices of $S \setminus \{x\}$. And as shown in [TV12], if a graph with no star star cutset has a non-path 2-join, then it has an extreme non-path 2-join. In Section 2.5 of [Cha+12] we showed how to find an extreme non-path 2-join in time $O(n^3m)$ in graphs that have no star cutset. It is in fact interesting that for *all* known algorithms that use 2-join detection (see the list below), one actually needs to look for a non-path 2-join in graphs that do not have star cutsets. This remark could perhaps lead to further speed ups.

Let us now summarize the results of [Cha+12] described above

Theorem 3.4.2 (*[Cha+12]*)

There exists :

- *an algorithm in time $O(n^2m)$ to find a 2-join (or output that it has none),*
- *an algorithm in time $O(n^2m)$ to find a non path 2-join (or output that it has none),*
- *an algorithm in time $O(n^3m)$ to find an extreme non path 2-join (or output that it has none) if the graph has no star cutset.*

The consequences are the following speed-ups of existing algorithms. Note that the speed-ups are sometimes more than by a factor of $O(n^2)$. This is because in the algorithms mentioned below, even cruder implementations of non-path 2-join detection are used.

1. Detecting the existence of a balanced skew partitions in Berge graphs in time $O(n^5)$ instead of $O(n^9)$ [Tro08].

2. The decomposition based recognition algorithm for Berge graphs in [Chu+05a] is now $O(n^{15})$ instead of $O(n^{18})$ (which is of moderate interest since the recognition algorithm in the same paper that is not based on the decomposition method is $O(n^9)$).

3. Finding a maximum weighted clique and a maximum weighted stable set in time $O(n^6)$ instead of $O(n^9)$ in Berge graphs with no balanced skew partition and no homogeneous pairs, and finding an optimal colouring in time $O(n^7)$ instead of $O(n^{10})$ for the same class [TV12].

4. Finding a maximum weighted stable set in time $O(n^6)$ instead of $O(n^9)$ in even-hole-free graphs with no star cutset [TV12].

As far as we care only for these applications, it is not immediately usable to try detecting non-path 2-joins faster than $O(n^2m)$, because $O(n^5)$ is a bottleneck independent from 2-join detection for all the algorithms mentioned here. An $O(n^4)$-time algorithm for extreme (or minimally-sided) non-path 2-joins would allow a speed-up of a factor n in the algorithms 3 and 4. We leave this as an open question.

As explained before, one of the major open algorithmic problems in the perfect graphs area is the existence of a purely combinatorial algorithm to colour Berge Graphs. It is known that one can obtain an optimal colouring of a perfect graph in polynomial time due to the algorithm of Grötschel, Lovász and Schrijver [GLS81]. This algorithm however is not purely combinatorial and is usually considered impractical. In [TV12], Trotignon and Vušković used the decomposition theorem of [Chu06] to devise a polynomial algorithm for colouring perfect graphs with no balanced skew partition, no homogeneous pair, nor complement 2-join. This was then improved in [Chu+15], Chudnovsky et al. propose an algorithm for perfect graphs with no balanced skew partition. In this paper, the authors have to deal with 2-joins and thus make use of the algorithms of [Cha+12] described above to detect non path 2-join and generalize these algorithms for trigraphs. Finally this result was used in [Chu+17a] where the authors prove a polynomial algorithm to colour perfect graphs with bounded clique number (not FPT though : the algorithm has complexity $O(n^{(\omega(G)+1)^2})$).

Chapter 4

Searching and Ordering

4.1 Introduction

Graph Searching is a strategy that permits to state structural theorems as well as design efficient algorithms. The idea is the following : one sequentially explores the vertices of a graph with a certain set of rules to do so. The algorithm produces a linear order on the vertices of the input graph, and there are several theorems that are of the kind : if the graph belongs to such class $\mathcal{C}$, then this search algorithm with produce an order of that particular type.

Moreover, even if one has in hand a deep decomposition theorem for an hereditary class $\mathcal{C}$ (like the one for perfect graphs, or even-hole free graphs, see surveys [Tro13] and [Vuš10]), these may sometimes be hard to use for algorithmic purposes if one cannot deal with one of the decompositions appearing in the theorem. On the opposite hand, just the existence of a vertex with some local structural property is sometimes enough for the design of efficient algorithms. Even for chordal graphs that are rather well structured, elimination orderings are the basis for the fastest algorithms.

One particular search algorithm called *LexBFS* (for Lexicographic Breadth First Search), due to Rose, Tarjan and Lueker [RTL76], that computes in linear time an ordering of the vertices of an input general graph, has attracted a lot of attention. In the case of chordal graphs, the ordering it produces is a perfect elimination ordering. This therefore provides us with an efficient algorithm to recognize chordal graphs, as if at one step one visits a vertex which is not simplicial with respect to what is before, one knows it is not a chordal graph, and if it never does, then the simplicial ordering is a certificate. It was also used for a linear time recognition for cographs by Bretscher, Corneil and Habib [Bre+03] and to compute diameters of graph. Almost all of this chapter relates to LexBFS, by exposing some results obtained with various coauthors on the properties and applications of this algorithm. Below is an overview of the contents.

- A first section will be devoted to the the algorithm Lex-BFS itself. We expose some general properties of LexBFS that will be of use in the following subsections.
- The following section will expose results published in [Abo+15] about applications of Lex-BFS to algorithmic questions around some hereditary classes, notably some generalizations of chordal graphs.

- In the next section we will discuss two results about questions related to these search algorithms on various classes of graphs. One concerns what happens when one does several iterations of the LexBFS algorithms, each one starting with the last vertex visited in the previous one. This contain results from [Cha+17].The other addresses the problem EndVertex which is the algorithmic question of deciding whether an input vertex can be the last in the order produced by different search algorithms (including LexBFS). We will see that this problem can be either polynomial or NP-hard depending on the search and/or the class considered (these results come from [CHM14]).

4.2 Lex BFS : definitions and properties

We present formally LexBFS in Algorithm 1 below. It is a variant of Breadth First Search 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 $\{0, ..., n-1\}$. By convention ϵ denotes the empty word. 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 $\text{label}(s) \leftarrow \{n+1\}$
for $i \leftarrow 1$ to **n do**
  pick an unnumbered vertex v with lexicographically largest label
  $\sigma(i) \leftarrow v$ ▷ v is assigned the number i
  for each unnumbered vertex w adjacent to v **do**
    append$(n-i)$ to $\text{label}(w)$
  end for
end for

One important fact is that LexBFS can be implemented to run in $O(m)$ time using partition refinement [Hab+00]. A linear order on the vertices produced by an execution of LexBFS is called a LexBFS order. One nice result is that one can characterize easily such an order, as noted in [BDN97].

Theorem 4.2.1 (*Brandstädt, Dragan and Nicolai [BDN97]*)

An ordering $\prec$ of the vertices of a graph $G = (V, E)$ is a LexBFS ordering if and only if it satisfies the following property: for all $a, b, c \in V$ such that $c \prec b \prec a$, $ca \in E$ and $cb \notin E$, there exists a vertex d in G such that $d \prec c$, $db \in E$ and $da \notin E$.

With this characterization, an easy but important fact about LexBFS orderings appears clearly : if $(v_1, v_2, \ldots, v_n)$ is a LexBFS ordering of G, then for all i $(v_1, v_2, \ldots, v_i)$ is a LexBFS ordering of $G[v_1, v_2, \ldots, v_i]$. Therefore if for a given class of graphs we are able to prove that the last vertex of a LexBFS has a certain property, then this property will be true for every vertex in the graph induced by its predecessors. Typically if you prove that the last vertex has a $\mathcal{F}$-free neighbourhood, it implies that any LexBFS ordering is an $\mathcal{F}$-elimination ordering. This will be crucial in the next section results.

Let us mention here that this so called ”4-points characterization” can be extended (as proved in [CK08]) for general BFS and DFS (Depth First Search) and its variant LexDFS. We recall it here for completeness and because the symmetry of these characterizations is very nice. (LexDFS was defined by Corneil and Krueger in [CK08], and has had since that several applications [Cor+16; CDH13; MC12] in particular to cocomparability graphs. Note that contrary to LexBFS no linear time implementation is known for computing a LexDFS ordering for general graphs (but there is one for cocomparability graphs, see [KM14]).)

Theorem 4.2.2 (*Corneil and Krueger*)

Let $G = (V, E)$ be a graph. Let $\prec$ be a total order on the vertices of G. Define a triple of vertices (a, b, c) to be a characteristic triple if $c \prec b \prec a$, $ca \in E$ and $cb \notin E$

- *$\prec$ is a BFS-ordering if and only if for every characteristic triple (a, b, c), there exists d such that $d \prec c$, $db \in E$.*
- *$\prec$ is an LBFS-ordering if and only if for every characteristic triple (a, b, c), there exists d such that $d \prec c$, $db \in E$ and $da \notin E$.*
- *$\prec$ is a DFS-ordering if and only if for every characteristic triple (a, b, c), there exists d such that $c \prec d \prec b$ and $db \in E$.*
- *$\prec$ is an LDFS-ordering if and only if for every characteristic triple (a, b, c), there exists d such that $c \prec d \prec b$, $db \in E$ and $da \notin E$.*

Let us also finish by giving the now classical proof that LexBFS orderings are simplicial orderings in the case of chordal graphs using the 4 point characterization. Let a be the last vertex in a LexBFS ordering $\prec$. Assume by contradiction that a is not simplicial and therefore has neighbours $a_3 \prec a_2$ that are not adjacent one to another. Assume that a_3 and a_2 are chosen minimal (with respect to $\prec$) for this property. Then we are exactly in the setting of Theorem 4.2.1 $(a, b, c) = (a, a_2, a_3)$ and thus there exists $a_4 \prec a_3$ (and we also chose it minimal) that is a neighbour of a_2 and not of a. But then a_3a_4 cannot be an edge, or else we would have an induced C_4. So it is not an edge and we are again in the setting of Theorem 4.2.1 for $(a, b, c) = (a_2, a_3, a_4)$ and get a vertex $a_5 \prec a_4$ that is a neighbour of a_3 and not of a_2 (and not of a because of the minimality of a_3). Again it cannot be a neighbour of a_4 because it would create a 5-hole, so the previous argument can be repeated again an again. Since the number of vertices is finite, we get a contradiction.

4.3 Elimination Orderings obtained by LexBFS and Applications

In this section we expose the results of [Abo+15]. We begin by proving a general tool result about elimination orderings given by LexBFS when the graph satisfies a local property called locally decomposable. We also define graphs called Truemper configurations that appear in the next two subsections.

In the following subsection we give two classes of graphs for which the existence of an $\mathcal{F}$-elimination ordering was already known (namely even-hole-free graphs and square-theta-free Berge graphs). We explain for each of them how our tool could be used to prove the existence of the ordering. For even-hole-free graphs, our method leads to speeding up the algorithm that computes a maximum clique. To be more specific, it turns out that these classes are slight generalizations of even-hole-free graphs and square-theta-free Berge graphs, defined by excluding different Truemper configurations, that are special types of graphs that play an important role in the study of hereditary graph classes (see survey [Vuš13]).

Then we apply systematically our method to produce classes of graphs that admit $\mathcal{F}$-elimination orderings for all possible non-empty sets of graphs $\mathcal{F}$ made of non-complete graphs on three vertices (there are seven such sets $\mathcal{F}$). This leads us to define seven classes of graphs, each of which having its own elimination ordering by our method. Two of these classes were previously studied (namely universally signable graphs and wheel-free graphs) and five of them were new. For almost all these classes, we got something from the ordering: a bound on the chromatic number, a colouring algorithm, or an algorithm for the maximum clique problem. Surprisingly, this systematic application of the method leads again to classes that are all defined by excluding some Truemper configurations.

We now sum up the previously known optimization algorithms for which we get better complexity (each time, we improve the previously known complexity by at least a factor of n):

- Maximum weighted clique in even-hole-free graphs in time $O(nm)$.
- Maximum weighted clique in universally signable graphs in time $O(n + m)$.
- Colouring in universally signable graphs in time $O(n + m)$.

4.3.1 A Theorem on LexBFS and a Local Property

As pointed out before Theorem 4.2.1 implies easily that for chordal graphs LexBFS orders are indeed simplicial orders. In [Abo+15] we wanted to extend this fact (and the algorithmic consequences of it) to generalizations of chordal graphs. To do that we proved the following theorem, using the 4 points characterization of Theorem 4.2.1

Theorem 4.3.1 (*[Abo+15]*)

Assume graph G is not a clique, and let z be the last vertex of a LexBFS ordering of G. Then there exists a connected component C of $G \setminus N[z]$ such that for every neighbour x of z,

- *either* $N(x) = N(z) \cup \{z\}$,
- *or* $N(x) \cap C \neq \emptyset$.

Equivalently, if we put together z with its neighbours of the first type, the resultant set of vertices is a clique, a module, and its neighbourhood is a minimal separator. (These are called moplexes in [BB98], and this theorem could be deduced from the results in this paper, but our proof using 4 points condition seemed more elegant and efficient).

Locally $\mathcal{F}$-decomposable class By definition, to be locally $\mathcal{F}$-free (i.e. every neighbourhood is $\mathcal{F}$-free) is a stronger property than to have an $\mathcal{F}$-elimination order (every induced subgraph of G has one vertex with $\mathcal{F}$-free neighbourhood). As we will explain shortly after, the property of being *locally $\mathcal{F}$- decomposable*, defined below, sits between those two (note that it was first introduced by Maffray, Trotignon and Vušković in [MTV08], under the name Property $(\star)$). If F is a subgraph of G, a vertex v is said to be *F-universal* if F is a subset of $N(v)$. Now if $\mathcal{F}$ is a family of graphs, a graph G is *locally $\mathcal{F}$- decomposable* if for every vertex v of G, every $F \in \mathcal{F}$ contained in $N(v)$, and every C connected component of $G \setminus N[v]$, there exists $y \in F$ such that y is not F-universal and $y \notin N(C)$. A class of graphs $\mathcal{C}$ is *locally $\mathcal{F}$-decomposable* if every graph $G \in \mathcal{C}$ is locally $\mathcal{F}$-decomposable.

It is easy to see that this is a hereditary property. Note also that it implies that the neighbourhood of any such component C is $\mathcal{F}$-free, so as long as the graph is not a clique, this property implies directly the existence of a $\mathcal{F}$-free cutset. In fact, we have more, since there always exists a vertex with a $\mathcal{F}$-free neighbourhood by the following theorem, which is a direct consequence of Theorem 4.3.1.

Theorem 4.3.2 (*[Abo+15]*)

If G is not a clique and is locally $\mathcal{F}$-decomposable then every LexBFS ordering of G is an $\mathcal{F}$-elimination ordering.

4.3.2 Truemper configurations

Let us present here what are called Truemper configurations, a notion that will be needed in the two next subsections. A *3-path configuration* is a graph induced by three internally vertex disjoint paths of length at least 1, $P_1 = x_1 \dots y_1$, $P_2 = x_2 \dots y_2$ and $P_3 = x_3 \dots y_3$, such that either $x_1 = x_2 = x_3$ or x_1, x_2, x_3 are all distinct and pairwise adjacent, and either $y_1 = y_2 = y_3$ or y_1, y_2, y_3 are all distinct and pairwise adjacent. Furthermore, the vertices of $P_i \cup P_j$, $i \neq j$, induce a hole. Note that this last condition in the definition implies the following.

- If x_1, x_2, x_3 are distinct (and therefore pairwise adjacent) and y_1, y_2, y_3 are distinct, then the three paths have length at least 1. In this case, the configuration is called a *prism*.
- If $x_1 = x_2 = x_3$ and $y_1 = y_2 = y_3$, then the three paths have length at least 2 (a path of length 1 would form a chord of the cycle formed by the two other paths). In this case, the configuration is called a *theta*.
- If $x_1 = x_2 = x_3$ and y_1, y_2, y_3 are distinct, or if x_1, x_2, x_3 are distinct and $y_1 = y_2 = y_3$, then at most one of the three paths has length 1, and the others have length at least 2. In this case, the configuration is called a *pyramid*.

A *wheel* (H, v) is a graph formed by a hole H, called the *rim*, and a vertex v, called the *centre*, such that the centre has at least three neighbours on the rim. A *Truemper configuration* is a graph that is either a prism, a theta, a pyramid or a wheel (see Figure 4.1). Since every Truemper configuration contains a hole, any class of graphs defined by excluding some Truemper configurations is a generalization of chordal graphs. Truemper configurations play a central role in the most studied hereditary classes, such as perfect graphs, claw-free graphs and even-hole-free graphs (see the next subsection and the survey [Vuš13] for more on this).

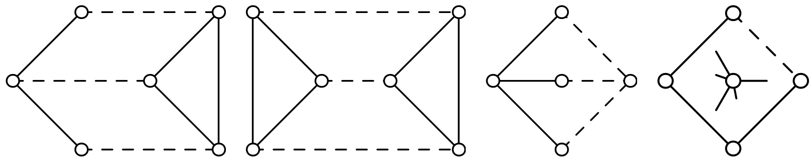


Figure 4.1: Pyramid, prism, theta and wheel (dashed lines represent paths)

4.3.3 Even-hole-free graphs and perfect graphs

In this subsection, we show how local decomposability can be used to provide elimination orderings and algorithms for even-hole-free graphs and some Berge graphs. The structural essence of odd-hole-free and even-hole-free graphs is captured by their generalizations to signed graphs. A graph is *odd-signable* if there exists an assignment of $0, 1$ weights to its edges that makes every chordless cycle of odd weight. A graph is *even-signable* if there exists an assignment of $0, 1$ weights to its edges that makes every triangle of odd weight and every chordless cycle of even weight. In [Tru82] Truemper proved a theorem that characterizes graphs whose edges can be assigned $0, 1$ weights so that chordless cycles have prescribed parities. The characterization states that this can be done for a graph G if and only if it can be done for all Truemper configurations contained in G. An easy consequence of this theorem when applied to odd-signable and even-signable graphs gives the following characterizations of these classes (see [Con+99]). A *sector* of a wheel is a subpath of the rim of length at least 1 whose ends are adjacent to the center and whose internal vertices are not. A wheel is *even* if it has an even number of sectors, and it is *odd* if it has an odd number of sectors of length 1.

- A graph is *odd-signable* if and only if is (theta, prism, even wheel)-free.
- A graph is *even-signable* if and only if it is (pyramid, odd wheel)-free.

Let us now show two results on vertex elimination orderings using local $\mathcal{F}$-decomposability. These results were known already (see [DV07] and [MTV08]), and were obtained by a special kind of lexicographic ordering of the vertices that is different from LexBFS (but more closely related to decomposition). Proving the existence of the ordering directly from Theorem 4.3.2 allows in both cases for the desired ordering to be computed in linear-time.

Theorem 4.3.3 (*da Silva and Vušković [DV07]*)

4-hole-free odd-signable graphs are locally hole-decomposable.

Note that 4-hole free odd-signable graphs generalize even hole-free graphs. The above theorem and Theorem 4.3.2 together imply that 4-hole-free odd-signable graphs admit a hole-elimination ordering. Theorem 4.3.3 is used in [DV07] to obtain a robust $O(n^2m)$-time algorithm for computing a maximum weighted clique in a 4-hole-free odd-signable graph (and hence in an even-hole-free graph). We showed how to reduce this complexity to $O(nm)$.

Theorem 4.3.4 (*[Abo+15]*)

There is an $O(nm)$-time algorithm whose input is a weighted graph G and whose output is a *maximum weighted clique of G or a certificate proving that G is not 4-hole-free odd-signable.*

The second application concern Berge graphs (or more precisely even-signable graphs that generalize them). A *square-theta* is a theta that contains a 4-hole. A *long hole* is a hole of length at least 5. Again a locally decomposable theorem was known.

Theorem 4.3.5 (*Maffray, Trotignon and Vušković [MTV08]*)

Square-theta-free even-signable graphs are locally long-hole-decomposable.

Again, combined with Theorem 4.3.2, we immediately get that square-theta-free even-signable graphs admit a long-hole-elimination ordering.

4.3.4 Some Generalizations of Chordal Graphs

Here we apply systematically our method to all possible sets made of non-complete graphs of order 3. This leads to seven classes of graphs, two of which were studied before (namely universally signable graphs and wheel-free graphs).

To describe the classes of graphs that we obtain, we need to be more specific about wheels. A wheel is a *1-wheel* if for some consecutive vertices x, y, z of the rim, the center is adjacent to y and non-adjacent to x and z. A wheel is a *2-wheel* if for some consecutive vertices x, y, z of the rim, the center is adjacent to x and y, and non-adjacent to z. A wheel is a *3-wheel* if for some consecutive vertices x, y, z of the rim, the center is adjacent to x, y and z. Observe that a wheel can be simultaneously a 1-wheel, a 2-wheel and a 3-wheel. On the other hand, every wheel is a 1-wheel, a 2-wheel or a 3-wheel. Also, any 3-wheel is either a 2-wheel or a *universal wheel* (that is a wheel whose center is adjacent to all vertices of the rim).

Up to isomorphism, there are four graphs on three vertices, and three of them are not complete, namely the independent graph on three vertices denoted by S_3, the path of length 2 denoted by P_3 and its complement denoted by $\overline{P_3}$).

Table 4.1 describes eight different classes of graphs $\mathcal{C}_1, \dots, \mathcal{C}_8$, all defined by excluding induced subgraphs described in the second column of the table. The third column describes a class $\mathcal{F}_i$ and the last column describes the class of $\mathcal{F}_i$-free graphs. We proved the following

Theorem 4.3.6

For $i = 1, \dots, 8$, let $\mathcal{C}_i$ and $\mathcal{F}_i$ be the classes defined as in Table 4.1. Then the C_i is exactly the class of locally $\mathcal{F}_i$-decomposable graphs.

Inclusions between these classes and several known classes are represented in Figure 4.2 (where the *diamond* is the graph obtained from K_4 by removing one edge, a *cap* is cycle of length at least 5 with a unique chord joining two vertices at distance 2 on the cycle, a *d-hole* is a 3-wheel such that the center has degree 3, and the *claw* is $K_{1,3}$). Observe that a d-hole is also a 2-wheel.

With Theorem 4.3.2, this directly implies the following.

Theorem 4.3.7

For $i = 1, \dots, 8$, let $\mathcal{C}_i$ and $\mathcal{F}_i$ be the classes defined as in Table 4.1. Then every LexBFS ordering of a graph of $\mathcal{C}_i$ is an $\mathcal{F}_i$-elimination ordering.

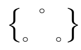
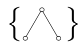
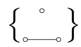
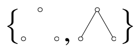
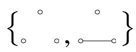
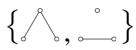
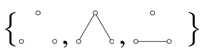
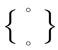
i	Class $\mathcal{C}_i$	$\mathcal{F}_i$	$\mathcal{F}_i$-free
1	(1-wheel, theta, pyramid)-free		no stable set of size 3
2	3-wheel-free		disjoint union of cliques
3	(2-wheel, prism, pyramid)-free		complete multipartite
4	(1-wheel, 3-wheel, theta, pyramid)-free		disjoint union of at most two cliques
5	(1-wheel, 2-wheel, prism, theta, pyramid)-free		stable sets of size at most 2 with all possible edges between them
6	(2-wheel, 3-wheel, prism, pyramid)-free		clique or stable set
7	(wheel, prism, theta, pyramid)-free		clique or stable set of size 2
8	hole-free		clique

Table 4.1: Eight classes of graphs

Already Known Classes We now describe the two classes of graphs from Table 4.1 that (apart from $\mathcal{C}_8$ = chordal graphs) were studied before. The first one is $\mathcal{C}_7$, i.e. graphs that contain no Truemper configuration, or equivalently by Theorem 4.3.7, graphs that are $\mathcal{F}_7$-locally decomposable. These are studied in [Con+97], where they are called *universally signable graphs*. The existence of a vertex whose neighbourhood is $\mathcal{F}_7$-free given by Theorem 4.3.7 is exactly the following theorem from [Con+97], that was originally proved through a global decomposition theorem. Theorem 4.3.7 provides a shorter proof as well as an algorithm that outputs the ordering that does not rely on global decomposition. In the next subsection, we study several algorithmic consequences.

Theorem 4.3.8 (*Conforti, Cornuéjols, Kapoor and Vušković [Con+97]*)

Every non-empty universally signable graph contains a simplicial vertex or a vertex of degree 2.

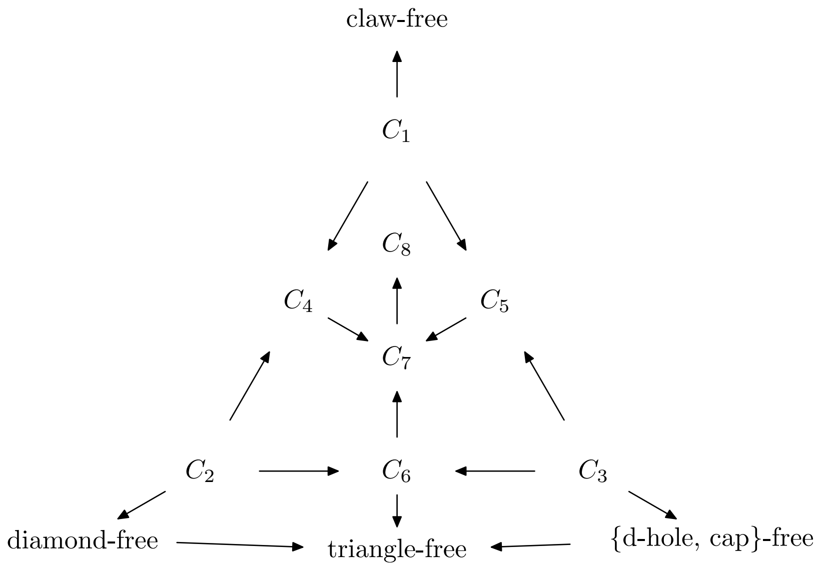


Figure 4.2: Inclusion for several classes of graphs. An arrow from A to B means "A is contained in B". Arrows arising from transitivity are not represented.

The second class that was studied previously is the class of wheel-free graphs and its superclass $\mathcal{C}_2$. These might have interesting structural properties, as suggested by several subclasses, see [Abo+12] for example for a list of them. The next theorem (which follows from Theorem 4.3.7 for $i = 2$) states the only non-trivial property that is known to be satisfied by all wheel-free graphs. The original proof (due to Chudnovsky who communicated it to us but did not publish it) is by induction, and the proof relying on our method is much shorter.

Theorem 4.3.9

Every non-empty 3-wheel-free graph contains a vertex whose neighbourhood is a disjoint union of cliques.

The following extends a well-known fact: a chordal graph G has at most n maximal cliques.

Corollary 4.3.10

A 3-wheel-free graph G has at most m maximal cliques.

Algorithmic consequences Table 4.2 describes several properties of the classes defined in Table 4.1. We indicate a reference for the properties that are already known, or follow easily from the given references.

Let us analyze the column "Max clique" of Table 4.2, that gives the best complexity of finding a maximum weighted clique in a graph of the corresponding class. By a result of Poljak [Pol74], it is NP-hard to compute a maximum stable set in a triangle-free graph. Rephrased in the complement, it is NP-hard to compute a maximum clique in an S_3-free graph, and therefore in graphs from $\mathcal{C}_1$. Finding a maximum weighted clique in $\mathcal{C}_2$ is easy as follows: for every vertex v, look for a maximum

i	Max clique	colouring
1	NP-hard [Pol74]	NP-hard [Hol81]
2	$O(nm)$ [RTL76]	NP-hard [MP96]
3	$O(nm)$	NP-hard [MP96]
4	$O(n+m)$	?
5	$O(nm)$	?
6	$O(n+m)$	NP-hard [MP96]
7	$O(n+m)$	$O(n+m)$
8	$O(n+m)$ [RTL76]	$O(n+m)$ [RTL76]

Table 4.2: Several properties of classes defined in Table 4.1

weighted clique in $N(v)$, and choose the best clique among these. This can be implemented by running n times the $O(n+m)$ algorithm of Rose, Tarjan and Lueker, because $N(v)$ is chordal for every v. In fact, this algorithm works in the larger class of universal-wheel-free graphs.

For $\mathcal{C}_4$, we need to be careful about the complexity analysis. Here is an algorithm that finds a maximum (weighted) clique in $G \in \mathcal{C}_4$. First by Theorem 4.3.7, we find in linear time an $\{S_3, P_3\}$-elimination ordering of G, say $(v_1, \dots, v_n)$. This means that in $G[\{v_1, \dots, v_i\}]$, $N(v_i)$ is a disjoint union of at most two cliques. We now show that, having this order, we can compute a maximum clique in time $O(m)$. We may assume that G is connected (otherwise we work on components separately), so $m \geq n-1$. Suppose inductively that a maximum clique of $G[\{v_1, \dots, v_{n-1}\}]$ is found in time $O(m - d(v_n))$. We now take the vertices of $N(v_n)$ one by one. We give name x and label X to first one, and check whether the next ones are adjacent to x. If so, we give them label X. If some are not adjacent to x, we give name y and label Y to the first one that we meet. The next vertices receive label X or Y according to their adjacency to x or y. Note that exactly one of these adjacencies must occur, since $N(v_n)$ is the union of at most two cliques. At the end of this loops, the vertices with label X and Y form at most two cliques in $N(v_n)$. They are identified in time $O(d(v_n))$. So, we now know all the maximal cliques of $G[N[v_n]]$ and a maximum clique of $G[\{v_1, \dots, v_{n-1}\}]$. A maximum clique among these is a maximum clique of G. All this takes time $O(m - d(v_n)) + O(d(v_n)) = O(m)$. Observe that this algorithm relies on a constant time checking of the adjacency, so it needs the graph to be represented by an adjacency matrix. Therefore, the time complexity is $O(n+m)$, but the space complexity is $O(n^2)$. Observe also that this algorithm is not robust. If the input graph is not in $\mathcal{C}_4$, the output is a set of vertices, and if it is a clique, we cannot be sure that it has maximum weight.

For class $\mathcal{C}_6$, the algorithm is similar to the previous one. We have to find a maximum clique in $N(v_n)$ in time $O(d(v_n))$. It is easy to verify quickly whether the neighbourhood of v_n is a clique or a stable set, and in both cases, it is immediate to find in time $O(d(v_n))$ a maximum weighted clique in it. We omit further details.

For $\mathcal{C}_3$ (that contains $\mathcal{C}_5$), the algorithm is similar to the previous one, except that we rely on a

$\{\overline{P_3}\}$-elimination ordering of G instead of an $\{S_3, P_3\}$-elimination ordering. As a result, the neighbourhood of the last vertex v is complete multipartite. We do not know how to find a maximum clique in $N(v)$ in time $O(d(v))$, so we do not know how to obtain a linear time algorithm. Instead, we look for a maximum clique in $N(v)$ in time $O(m)$, and therefore the overall complexity is $O(nm)$.

Let us now analyze the column "colouring" of Table 4.2, that gives the best complexity for colouring a graph of the corresponding class. Since the edge-colouring problem is NP-hard [Hol81], it follows that colouring line graphs is NP-hard, and therefore, so is colouring claw-free graphs (that are all in $\mathcal{C}_1$). Classes $\mathcal{C}_2, \mathcal{C}_3$ and $\mathcal{C}_6$ contain all triangle-free graphs, that are NP-hard to colour as proved by Preissmann and Maffray [MP96]. For $\mathcal{C}_7$, we first try to find a 2-colouring of the graph by the classical BFS algorithm. If it does not exist, we look for a $\max(3, \omega(G))$-colouring of the input graph G as follows. By Theorem 4.3.7 we obtain an $\{S_3, P_3, \overline{P_3}\}$-elimination ordering in linear time. As a result, the neighbourhood of the last vertex of the ordering is a clique or has size 2. We remove the last vertex v, colour recursively the remaining vertices, and give some available colour to v.

4.3.5 Open questions

Addario-Berry, Chudnovsky, Havet, Reed and Seymour [Add+08] proved that every even-hole-free graph admits a *bisimplicial* vertex - a vertex whose neighbourhood is the union of two cliques. The proof is difficult and it would be great if this result could be proved by some search algorithm like LexBFS.

Corollary 4.3.10 suggests that a linear time algorithm for the maximum clique problem might exists in $\mathcal{C}_2$, but we could not find it.

We are not aware of a polynomial time colouring algorithm for graphs in $\mathcal{C}_4$ or $\mathcal{C}_5$, but it would be surprising to us that it exists. A structural theorem and a polynomial time recognition algorithm was given recently [BRV] for the class $\mathcal{C}_4$.

Since class $\mathcal{C}_1$ generalizes claw-free graphs, it is natural to ask which of the properties of claw-free graphs it has, such as a structural description (see [CS08a]), a polynomial time algorithm for the maximum stable set (see [FOS11]), approximation algorithms for the chromatic number (see [KG09]), a polynomial time algorithm for the induced linkage problem (see [Fia+12]), and a polynomial χ-binding function (see [Gyá87]). Also we wonder whether theta-free graphs are χ-bounded by a *polynomial* (quadratic?) function (recall that in [KO04], they are proved to be χ-bounded). Recently Trotignon and Sintiari ([Tro18]) proved that the graphs in $\mathcal{C}_1$ have unbounded treewidth, even restricted to the triangle free case.

In [Con+97], an $O(nm)$ time algorithm is described for the maximum weighted stable set problem in $\mathcal{C}_7$. Since the class is a simple generalization of chordal graphs, we wonder whether a linear time algorithm exists.

4.4 Two Algorithmic Questions about LexBFS

4.4.1 The End Vertex Problem

As we explained before, studying the properties of the last vertex in a search algorithm like LexBFS can produce nice structural and algorithmic results for hereditary classes of graphs. This was the motivation, in [CHM14], to investigate the algorithmic question of deciding, with input a graph G and a vertex $v \in V(G)$, if v can be the last vertex of various searches for different classes of graphs.

In [CKL10], it was proved that it is NP-complete to determine if a given vertex could be last in a LexBFS-ordering. For basic searches such as BFS and DFS the question was left open. In Table 4.3, we summarize the knowledge about this question, and results in bold typeface are from [CHM14]. The proof of many NP-completeness results follow from the NP-completeness result for a subfamily of graphs. For example for DFS, we proved that the problem is NP-complete for strongly chordal split graphs, and from that it follows the same for all superclasses, like, split, chordal, weakly chordal and the class of all graphs. For unknown cases, we write in italic and between parenthesis our conjectures if any.

End-vertex results	BFS	LexBFS	DFS	LexDFS
All Graphs	**NPC**	NPC	**NPC**	**NPC**
- Bipartite	**NPC**	?(*NPC*)	?(*NPC*)	?(*NPC*)
- Weakly Chordal	**NPC**	NPC [CKL10]	**NPC**	**NPC**
- - Chordal	?(*NPC*)	?(*NPC*)	**NPC**	?
- - - Split	**P**	**P**	**NPC**	**P**
- - - - Str.Chordal Split	**P**	**P**	**NPC**	**P**
- - - Path Graphs	?	?(*P*)	**NPC**	?

Table 4.3

More general searches are MNS (maximum neighbourhood search) and MCS (maximum cardinality search) : in the first one the vertex chosen next must have a neighbourhood inside the previously visited vertices that is inclusion wise maximal (this is true in LexBFS or LexDFS) and in the second it has to be of maximum size (so any MCS order is in particular a MNS order). We conjectured in [CHM14] that MNS should be polynomial on all graphs, but recently Beisegel et al. [Bei+18] proved that it is in fact NP-complete, even for weakly chordal graphs (it was known to be polynomial for chordal graphs [Ber+10]). They also prove that MCS is NP-complete on all graphs.

The main open question that remains is the complexity on chordal graphs of the end-vertex problem for BFS, LexBFS, and LexDFS.

Moreover, in [CKL10], a very simple linear time algorithm for the end-vertex problem for LBFS on interval graphs is presented. We do not know if this extends to path graphs or cocomparability graphs (i.e. complements of comparability graphs). Since it is proved in [DH17] that for recognition LexBFS behaves the same on interval graphs and comparability graphs, we conjecture that it is polynomial on both classes. It could also be interesting to find a class of graphs for which BFS and LBFS behave differently for the end-vertex problem.

4.4.2 Iterating LexBFS and the 2-loop Conjecture

The $^+$ rule on LexBFS, introduced by Simon in [Sim91] and written as LexBFS^+, takes as input a graph $G = (V, E)$ and a total vertex ordering σ of G. $\text{LexBFS}^+(G, \sigma)$ is then the LexBFS vertex ordering of G obtained by first breaking ties normally using lexicographic labels, and if other ties remain among eligible vertices, chooses the eligible vertex that is *right most* in σ to visit next. In particular, one starts with the vertex that is last in σ.

For a given a graph G and order σ_0 on the vertices of G, computing a sequence of LexBFS searches on G: $\sigma_1, \sigma_2, \ldots$, where $\sigma_i = \text{LexBFS}^+(G, \sigma_{i-1})$, has been used to introduce fast recognition algorithms, known as *multisweep* algorithms, for graph families such as proper interval graphs, interval graphs and cocomparability graphs [Cor04; COS09; DH17]. Among such results is a recent result of [DH17] which states:

Theorem 4.4.1 (*Dusart and Habib, [DH17]*)

If G is a cocomparability graph G, σ_0 any order on the vertices of G and the sequence $\{\sigma_i\}_{i\geq 1}$ is defined by $\sigma_i = \text{LexBFS}^+(G, \sigma_{i-1})$, then σ_n where $n = |V(G)|$ is a transitive ordering of the complement of G.

Evidently, as the number of distinct orderings of vertices of a finite graph is finite, no matter which ordering σ_0 we start with, this sequence $\{\sigma_i\}$ of LexBFS^+ orderings will cycle eventually. That is, for some i and k, $\sigma_{i+k} = \sigma_i$. Thus if we keep running LexBFS^+ traversals, we will eventually loop. For general graphs there are two questions of interest on the subject:

- Among all possible choices of σ_0 as a start ordering, how long does it take to loop?
- How large can this cycle be?

Regarding the second question, and restricted to the class of cocomparability graphs, Dusart and Habib [DH17] have conjectured that the length of this largest cycle of vertex orderings is as small as possible:

Conjecture 4.4.2 (*2-loop Conjecture, Dusart and Habib, [DH17]*)

If G is a cocomparability graph G, σ_0 any order on the vertices of G and the sequence $\{\sigma_i\}_{i\geq 1}$ is defined by $\sigma_i = \text{LexBFS}^+(G, \sigma_{i-1})$, then there exists i such that $\sigma_i = \sigma_{i+2}$.

If we define $LexCycle(G)$ to be the maximal size of the cycle amongst all possible σ_0, the conjecture can be reformulated as : if G is a cocomparability graph then $LexCycle(G) = 2$.

In [Cha+17], we provided support for the conjecture of Dusart and Habib by proving the following.

Theorem 4.4.3

The 2-loop conjecture 4.4.2 is true for domino-free cocomparability graphs.

The domino is a graph obtained from a C_6 by joining a pair of antipodal vertices. Moreover, while this subclass of cocomparability graphs contains proper interval graphs, interval graphs, cographs and cobipartite graphs, for each of these cases, we gave an independent proof which provides stronger results, and sheds light into structural properties of these graph classes. Furthermore, we prove that the same bound of 2 holds for other graph classes: trees, which are not necessarily asteroidal triple-free graphs, as well as distance hereditary graphs.

A k-ladder is the graph represented on Figure 4.3.

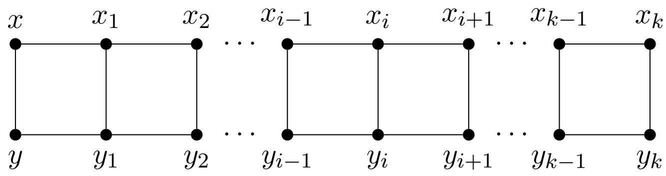


Figure 4.3: A k-ladder.

So a 1-ladder is a C_4 (so that 1-ladder free cocomparability graph are interval graphs) and a 2-ladder is a domino. Because of the way our proof for domino-free graphs works, we think that proving the 2-loop conjecture for k-ladder free graphs is the next step that should be considered.

A last remark on this problem. A well studied notion is the notion of *asteroidal set* : a set of vertices X such that for every distinct x, y, z in X, there exists a path linking y and z that avoids the neighbourhood of x. The maximal size of an asteroidal set is called *asteroidal number* of a graph G. The class of graphs with no asteroidal triple (AT-free graphs) is a large class of graphs that has been intensively studied, and it is not difficult to see that it contains cocomparability graphs. Stacho, [Sta] therefore asked if the 2-loop conjecture could be extended as $LexCycle(G) \leq an(G)$ for any graph. In [Cha+17], we answered Stacho's question negatively by exhibiting a graph G with $an(G) = 5$ but $LexCycle(G) \geq 12$. Nevertheless we still conjecture that $LexCycle(G) = 2$ for all AT-free graphs.

Chapter 5

Computing ω and α : beyond polynomiality

5.1 Introduction

In this chapter, we will focus on the problem of Maximum Independent Set and Maximum Clique for particular hereditary classes. Both problems are not only NP-Hard problems but also not approximable within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ unless $P = NP$ [Zuc07], and $W[1]$-hard [DF13]. Thus, it seems natural to study the complexity of these problems when restricted to certain graph classes and it is the subject of hundreds of papers. The first section is devoted to H-free graphs, and the second to Ball graphs. In both sections we will review the current knowledge on the problems and present some results on approximation or fixed parameter tractability that appeared respectively in [Bon+18b] and [Bon+18a].

5.2 H-free graphs

Note that since changing H to its complement transform one problem into another, we can choose one : here we state things in terms of α, so we study the problem of determining the size of a maximum independent set (MIS). For the sake of simplicity, "MIS" will denote the optimisation, decision and parameterized version of the problem (in the latter case, the parameter is the size of the solution), the correct use being clear from the context.

As stated before MIS in general graphs is a difficult problem. Unfortunately, excluding a graph H does not change much the difficulty as it turns out that for most graphs H, MIS in H-free graphs remains NP-hard, as shown by a very simple reduction first observed by Alekseev:

Theorem 5.2.1 (*[Ale82]*)

Let H be a connected graph which is neither a path nor a subdivision of the claw. Then MIS *is NP-hard in H-free graphs.*

On the positive side, the case of P_t-free graphs has attracted a lot of attention during the last decade. While it is still open whether there exists $t \in \mathbb{N}$ for which MIS is NP-hard in P_t-free graphs, quite involved polynomial-time algorithms were discovered for P_5-free graphs [LVV14], and very recently for P_6-free graphs [Grz+17]. In addition, we can also mention the recent following result: MIS admits a subexponential algorithm running in time $2^{O(\sqrt{tn \log n})}$ in P_t-free graphs for every $t \in \mathbb{N}$ [Bac+18].

The second open question concerns the subdivision of the claw. Let $S_{i,j,j}$ be a tree with exactly three vertices of degree one, being at distance i, j and k from the unique vertex of degree three. The complexity of MIS is still open in $S_{1,2,2}$-free graphs and $S_{1,1,3}$-free graphs. In this direction, the only positive results concern some subcases: it is polynomial-time solvable in $(S_{1,2,2}, S_{1,1,3}, dart)$-free graphs [Kar17], $(S_{1,1,3}, banner)$-free graphs and $(S_{1,1,3}, bull)$-free graphs [KM17], where $dart$, $banner$ and $bull$ are particular graphs on five vertices.

Given the large number of graphs H for which the problem remains NP-hard, it seems natural to investigate the existence of parameterized algorithms, that is, determining the existence of an independent set of size k in a graph with n vertices in time $O(f(k)n^c)$ for some computable function f and constant c. A very simple case concerns K_r-free graphs, that is, graphs excluding a clique of size r. In that case, Ramsey's theorem implies that every such graph G admits an independent set of size $n^{\frac{1}{r-1}}$, where n is the number of vertices of G. In the FPT vocabulary, it implies that MIS in K_r-free graphs has a kernel with k^{r-1} vertices.

To the best of our knowledge, the first step towards an extension of this observation within the FPT framework is the work of Dabrowski *et al.* [Dab+12] (see also Dabrowski's PhD manuscript [Dab12]) who showed that for any positive integer r, MAX WEIGHTED INDEPENDENT SET is FPT in H-free graphs when H is a clique of size r minus an edge. In the same paper, they settle the parameterized complexity of MIS on almost all the remaining cases of H-free graphs when H has at most four vertices. The conclusion is that the problem is FPT on those classes, except for $H = C_4$ which is left open. We answer this question by showing (among others) that MIS remains $W[1]$-hard in C_4-free graphs.

Finally, we can also mention the case where H is the *bull* graph, which is a triangle with a pending vertex attached to two different vertices. For that case, a polynomial Turing kernel was obtained [TTV17] then improved in [CS18].

The results of [Bon+18b]

- We present three reductions proving $W[1]$-hardness of MIS in graph excluding several graphs as induced subgraphs, such as $K_{1,4}$, any fixed cycle of length at least 4, and any fixed tree with two branching vertices. We propose a definition of a graph decomposition whose aim is to capture all graphs which can be excluded using our reductions.

- We also extend the polynomial algorithm of Alekseev when H is a disjoint union of edges to an FPT algorithm when H is a disjoint union of cliques. (We conjecture more generally, that the disjoint union of two easy cases is an easy case; formally, *if* MIS *is FPT in G-free graphs and in H-free graphs, then it is FPT in $G \uplus H$-free graphs*).

- We present a general framework extending the technique of *iterative expansion*, which itself is the maximization version of the well-known iterative compression technique. We apply this framework to provide FPT algorithms when H is a clique minus a complete bipartite graph, or when H is a clique minus a triangle, and when H is the gem graph (a P_4 with an additional vertex).

- Finally, we focus on the existence of polynomial (Turing) kernels. We first strengthen some results of the previous section by providing polynomial (Turing) kernels in the case where H is a clique minus a claw. Then, we prove that for many H, MIS on H-free graphs does not admit a polynomial kernel, unless $NP \subseteq coNP/poly$.

Our results allows to obtain the complete classification in terms of polynomial/polynomial kernel/no polynomial kernel but polynomial Turing kernel/$W[1]$-hard for all possible graphs on four vertices, while only five graphs on five vertices remain open for the FPT/$W[1]$-hard dichotomy.

5.3 Intersection Graphs of Balls

Most of the hard optimization and decision problems remain NP-hard on disk graphs and even unit disk graphs. For instance, disk graphs contain planar graphs [Koe36] on which several of those problems are intractable. However, shifting techniques and separator theorems may often lead to subexponential classic or parameterized algorithms [AF04; MP15; SW98; Bir+17]. Many approximation algorithms have been designed specifically on (unit) disk graphs, or more generally on geometric intersection graphs, see for instance [Cha03; NHK04; NH05; EJS05; Lee06; GP10] to cite only a few. Besides ad hoc techniques, local search and VC-dimension play an important role in the approximability of problems on (unit) disk graphs. For the main packing and covering problems (MAXIMUM INDEPENDENT SET, MIN VERTEX COVER, MINIMUM DOMINATING SET, MINIMUM HITTING SET, and their weighted variants) at least a PTAS is known.

However, all the techniques that we mentioned are only amenable to packing and covering problems. The MAXIMUM CLIQUE problem is arguably the most prominent problem which does not fall into those categories. For example, anything along the lines of exploiting a small separator cannot work for MAXIMUM CLIQUE, where the densest instances are the hardest. Therefore, it seems that new ideas are necessary to get improved approximate or exact algorithms for this problem. This is why, in this paper, we focus on solving MAXIMUM CLIQUE on (unit) disk graphs in dimension 2 or higher.

In 1990, Clark *et al.* [CCJ90] gave an elegant polynomial-time algorithm for MAXIMUM CLIQUE on unit disk graphs when the input is a geometric representation of the graph. It goes as follows: guess in quadratic time the two more distant centers of disks in a maximum clique (at distance at most 2), remove all the centers that would contradict this maximality, observe that the resulting graph is co-bipartite. Hence, one can find an optimum solution in polynomial time by looking for a maximum independent set in the complement graph, which is bipartite. However, recognizing unit disk graphs is NP-hard [BK98], and even $\exists\mathbb{R}$-complete [KM12]. In particular, if the input is the mere unit disk graph, one cannot expect to efficiently compute a geometric representation in order to run the previous algorithm. Raghavan and Spinrad showed how to overcome this issue and suggested a polynomial-time algorithm which does not require the geometric representation [RS03]. Their algorithm is a subtle *blind* reinterpretation of the algorithm by Clark *et al.* It solves MAXIMUM CLIQUE on a superclass of the unit disk graphs or correctly claims that the input is not a unit disk graph. Hence, it cannot be used to efficiently recognize unit disk graphs.

The complexity of MAXIMUM CLIQUE on general disk graphs is a notorious open question in computational geometry. On the one hand, no polynomial-time algorithm is known, even when the geometric representation is given. On the other hand, the NP-hardness of the problem has not been established, even when only the graph is given as input.

The piercing number of a collection of geometric objects is the minimum number of points that hit all the objects. It is known since the fifties (although the first published records of that result came later in the eighties) that the piercing number of pairwise intersecting disks is 4 [Sta81; Dan86]. An account of this story can be found in a recent paper by Har-Peled *et al.* [Har+18]. Ambühl and Wagner observed that this yields a 2-approximation for MAXIMUM CLIQUE [AW05]. Indeed, after guessing in polynomial time four points hitting a maximum clique and removing every disk not hit by those points, the instance is partitioned into four cliques; or equivalently, two co-bipartite graphs. One can then solve optimally each instance formed by one co-bipartite graph and return the larger solution of the two. This cannot give a solution more than twice smaller than the optimum. Since then, the problem has proved to be elusive with no new positive or negative results. The question on the complexity and further approximability of MAXIMUM CLIQUE on general disk graphs is considered as folklore [Ban+06], but was also explicitly mentioned as an open problem by Fishkin [Fis03], Ambühl and Wagner [AW05]. Cabello even asked if there is a 1.99-approximation for disk graphs with two sizes of radii [Cab15b; Cab15a]. Recently, Bonnet *et al.* [Bon+18c] showed that the disjoint union of two odd cycles is not the complement of a disk graph. From this result, they obtained a subexponential algorithm running in time $2^{\tilde{O}(n^{2/3})}$ for MAXIMUM CLIQUE on disk graphs, based on a win-win approach. They also got a QPTAS by calling a PTAS for MAXIMUM INDEPENDENT SET on graphs with sublinear odd cycle packing number due to Bock *et al.* [Boc+14], or branching on a low-degree vertex.

The results of [Bon+18a]

Our main contributions are twofold. The first is a randomized EPTAS (Efficient Polynomial-Time Approximation Scheme, that is, a PTAS in time $f(\varepsilon)n^{O(1)}$) for MAXIMUM INDEPENDENT SET on graphs of $\mathcal{X}(d,\beta,1)$. The class $\mathcal{X}(d,\beta,1)$ denotes the class of graphs whose neighbourhood hypergraph has VC-dimension at most d, independence number at least βn, and no disjoint union of two odd cycles as an induced subgraph. (The VC-dimension is a notion that was introduced by Vapnik and Chervonenkis in the seminal paper [VC15]. It is defined as the largest size of a shattered set, where a shattered set is a set of vertices X such that for every subset Y of X, there exists an hyperedge h with the property that $h \cap X = Y$).

Theorem 5.3.1

For any constants $d \in \mathbb{N}$, $0 < \beta \leqslant 1$, for every $0 < \varepsilon < 1$, there is a randomized $(1-\varepsilon)$-approximation algorithm running in time $2^{\tilde{O}(1/\varepsilon^3)}n^{O(1)}$ for MAXIMUM INDEPENDENT SET *on graphs of $\mathcal{X}(d,\beta,1)$ with n vertices.*

Using the forbidden induced subgraph result of Bonnet *et al.* [Bon+18c], it is then easy to reduce MAXIMUM CLIQUE on disk graphs to MAXIMUM INDEPENDENT SET on $\mathcal{X}(4,\beta,1)$ for some constant β. We therefore obtain a randomized EPTAS (and a PTAS) for MAXIMUM CLIQUE on disk graphs, settling almost (the NP-hardness, ruling out a 1-approximation, is still to show) completely the approximability of this problem.

Theorem 5.3.2

There is a randomized EPTAS for MAXIMUM CLIQUE *on disk graphs, even without geometric representation. Its running time is* $2^{\tilde{O}(1/\varepsilon^3)}n^{O(1)}$ *for a* $(1-\varepsilon)$*-approximation on a graph with* n *vertices.*

The second contribution is to show the same forbidden induced subgraph for unit ball graphs as the one obtained for disk graphs. The proofs are radically different and the classes are incomparable. So the fact that the same obstruction applies for disk graphs and unit ball graphs might be somewhat accidental.

Theorem 5.3.3

A complement of a unit ball graph cannot have a disjoint union of two odd cycles as an induced subgraph. In other words, if G *is a unit ball graph, then* $\text{iocp}(\overline{G}) \leqslant 1$*.*

In the previous statement iocp denotes the *induced odd cycle packing number* of a graph, i.e., the maximum number of odd cycles as a disjoint union in an induced subgraph. Again, Theorem 5.3.1 and Theorem 5.3.3 naturally lead to:

Theorem 5.3.4

There is a randomized EPTAS in time $2^{\tilde{O}(1/\varepsilon^3)}n^{O(1)}$ *for* MAXIMUM CLIQUE *on unit ball graphs, even without the geometric representation.*

Before that result, the best approximation factor was 2.553, due to Afshani and Chan [AC05]. In particular, even getting a 2-approximation algorithm (as for disk graphs) was open.

Finally we show that such an approximation scheme, even in subexponential time, is unlikely for ball graphs (that is, 3-dimensional disk graphs with arbitrary radii), and unit 4-dimensional disk graphs. Our lower bounds also imply NP-hardness. To the best of our knowledge, the NP-hardness of MAXIMUM CLIQUE on unit d-dimensional disk graphs was only known when d is superconstant ($d = \Omega(\log n)$) [AH08].

In the following paragraphs, we sketch the principal lines of the two main contributions of the paper.

EPTAS for MAXIMUM INDEPENDENT SET on $\mathcal{X}(d, \beta, 1)$ The first main result of this paper asserts that if a graph G satisfies that every two odd cycles are joined by an edge, the Vapnik-Chervonenkis dimension of the hypergraph of the neighbourhoods of G is bounded, and $\alpha(G)$ is at least a constant fraction of $|V(G)|$, then $\alpha(G)$ can be computed in polynomial time at any given precision. More precisely, we present in that case a randomized EPTAS running in time $2^{\tilde{O}(1/\varepsilon^3)}n^{O(1)}$ and a deterministic PTAS.

Our algorithm works as follows. We start by sampling a small subset of vertices. Hoping that this small subset is entirely contained in a fixed optimum solution I, we include the selected vertices to our solution and remove their neighbourhood from the graph. Due to the classic result of Haussler and Welzl [HW86] on ε-nets of size $O(d/\varepsilon \log 1/\varepsilon)$ (where d is the VC-dimension), this sampling lowers the degree in I of the remaining vertices. We compute a shortest odd cycle. If this cycle is short, we can remove its neighbourhood from the graph and solve optimally the problem in the resulting graph, which is bipartite by assumption. If this cycle is long, we can efficiently find a small odd-cycle transversal. This is shown by a careful analysis on the successive neighbourhoods of the cycle, and the recurrent fact that this cycle is a shortest among the ones of odd length.

The complement of the union of two odd cycles is not a unit ball graph Given a needle in $\mathbb{R}^3$ whose middlepoint is attached to the origin, one can apply a continuous motion in order to turn it around (a motion à la Kakeya, henceforth *Kakeya motion*). A Kakeya motion can be seen as a closed antipodal curve on the 2-sphere. If we now consider two needles, each with a Kakeya motion, then the two needles have to go through a same position. This simply follows from the fact that two antipodal curves on the 2-sphere intersect. The second main result of this paper is a translation of this Jordan-type theorem in terms of intersection graphs: The complement of a unit ball graph does not contain the disjoint union of two odd cycles. The proof can really be seen as two Kakeya motions, each one along the two odd cycles, leading to a contradiction when the needles achieve parallel directions.

Together with the first result, it implies a randomized EPTAS for MAXIMUM CLIQUE on disk graphs, and for the following problem: Given a set S of points in $\mathbb{R}^3$, find a largest subset of S of diameter at most 1.

Remarks and further directions First, it is not difficult to modify our algorithms to address similarly the MAXIMUM WEIGHTED INDEPENDENT SET, the problem of finding a independent set of maximum weight in a graph where vertices are given weights.

One might wonder what is the constant hidden in $O(1)$ in the time complexity of the randomized EPTAS $f(\varepsilon)n^{O(1)}$. In our paper we showed how to achieve near quadratic time $f(\varepsilon)n^2 \log n$ where n is the number of vertices of our *unit ball graph G* (the geometric representation is not required).

The obvious remaining question is the complexity of MAXIMUM CLIQUE in disk graphs and in unit ball graphs. An interesting direction would be to find a toy problem on which we could prove NP-hardness. A nice class, which appears to be a subclass of unit ball graphs, is that of the so-called *Borsuk graphs*: We are given some (small) real $\varepsilon > 0$ and a finite collection V of unit vectors in $\mathbb{R}^3$. The Borsuk graph $B(V, \varepsilon)$ has vertex set V and its edges are all pairs $\{v, v'\}$ whose dot product is at most $-1 + \varepsilon$ (i.e. near antipodal). The difficulty of computing the (weighted) independence number on Borsuk graphs is also an open question. A notable subclass of Borsuk graphs where this problem is polynomial-time solvable is the class of the quadrangulations of the projective plane. These well-studied objects have the striking property to be either bipartite or 4-chromatic. Furthermore, the odd cycle packing number of these graphs is at most 1. Artmann et al. recently showed that so-called *bimodular integer programming*, that is integer programming where the constraint matrix has full rank and all its subdeterminants are in $\{-2, -1, 0, 1, 2\}$, can be solved in strongly polynomial time [AWZ17]. They also observe that MAXIMUM WEIGHTED INDEPENDENT SET on graphs with ocp $\geqslant 1$ is a bimodular integer programming problem. This implies the tractability of computing the weighted independence number on quadrangulations of the projective plane.

Another natural question is to find a superclass of geometric intersection graphs which both contain unit balls and disks. More generally, is it possible to explain why we have the same forbidden induced subgraph (the complement of a disjoint union of two odd cycles) for disk graphs and unit ball graphs? As already said the proofs of these two facts are quite different.

Let us call *quasi unit disk graphs* those disk graphs that can be realized for any $\varepsilon > 0$ with disks having all the radii in the interval $[1, 1 + \varepsilon]$. We showed that, for the clique problem, quasi unit ball graphs are unlikely to have a QPTAS, while unit ball graphs admit an EPTAS. In dimension 2, it can be easily shown that unit disk graphs form a proper subset of quasi unit disk graphs, which form themselves a proper subset of disk graphs. Can we find for this intermediate class an efficient exact algorithm solving MAXIMUM CLIQUE?

Problem 5.3.5

Is there a polynomial-time algorithm for MAXIMUM CLIQUE *on quasi unit disk graphs?*

Our randomized EPTAS works for MAXIMUM INDEPENDENT SET under three hypotheses. While it is clear that we crucially need that iocp $\geqslant 1$ (or at least that iocp is constant), as far as we can tell, the boundedness of the VC-dimension and the fact that the solution is fairly large might not be required.

Problem 5.3.6

Is there a(n E)PTAS for MAXIMUM INDEPENDENT SET *on graphs without the union of two odd cycles as an induced subgraph?*

Atminas and Zamaraev [AZ18] showed that the complement of $K_2 + C_s$ is not a unit disk graph when s is odd (where K_2 is an edge and C_s is a cycle on s vertices). Is this obstruction enough to obtain an alternative polynomial-time algorithm for MAXIMUM CLIQUE on unit disk graphs?

Problem 5.3.7

Is MAXIMUM INDEPENDENT SET *solvable in polynomial-time on graphs excluding the union of an edge and an odd cycle as an induced subgraph?*

Chapter 6

Colouring

6.1 Introduction

The chromatic number $\chi(G)$ of a graph - the least number of parts in a partition of the vertex set into independent sets - is arguably the most intensively studied graph invariant in the history of graph theory. In this chapter, we want to understand which hereditary classes have bounded chromatic number and which do not. Since speaking about an hereditary class of graphs is the same as speaking about a family of forbidden induced subgraphs, the previous question of asking which hereditary classes have bounded chromatic number can also be stated the following way : which induced subgraphs can be guaranteed to exist in a graph that has arbitrarily large chromatic number? (It is not in the scope of this document, but note that the same question is also sensible for other graph containment relations than induced subgraphs. The similar question with minor relation instead includes one of the most famous open problem in graph theory : Hadwiger's Conjecture).

One obvious inequality about chromatic number of a given graph is that it is at least its clique number - the size of the largest complete subgraph it contains. But can we say that having very large chromatic number guarantees the existence of a large complete subgraph? The answer is well known to be no, as it is now a classical fact in graph theory that chromatic number can be arbitrarily far apart from clique number : there exists graphs that are triangle free that have arbitrarily large chromatic number, by classical constructions of Tutte (under the nom-de-plume Blanche Descartes) [Des54], Mycielski [Myc55], or Zykov [Zyk49]. Moreover things can be strengthened, since a celebrated result or Erdős ([Erd59]) we also know that for any k, there exists graphs such that the ball of radius k around each vertex induces a tree (or, equivalently, the graph does not contain any cycle of length at most $2k+1$), that have chromatic number at least k. More precisely (recall that the girth of a graph is defined as the size of its smallest cycle).

Theorem 6.1.1 (*Erdős, [Erd59]*)

For any integers k and l, there exists a graph G such that $\chi(G) \geq k$ and $\mathrm{girth}(G) \geq l$.

Note that this means in particular that one should think of chromatic number as a global notion and not a local one, as a graph can locally look like a 2-chromatic graph (a tree) and still have arbitrarily large chromatic number.

Since most studied classes in fact contain cliques or arbitrary size, and hence do not have bounded chromatic number, it makes sense to ask whether when one restricts a class to its members of bounded clique number, one gets a class with bounded chromatic number. This is the meaning of the following notions, introduced by Gyárfás [Gyá87].

Definition 6.1.2

- *A class* $\mathcal{C}$ *of graphs is* chi-bounded *if*

$$\exists f_{\mathcal{C}} : \mathbb{N} \to \mathbb{N}, \text{ such that } \forall G \in \mathcal{C}, \chi(G) \leq f_{\mathcal{C}}(\omega(G))$$

- *A family* $\mathcal{F}$ *of graphs is* chi-bounding *if* $\mathrm{Forb}(\mathcal{F})$ *is chi-bounded.*

For example, the class of all graphs is of course not chi-bounded, and neither is the class of C_4-free graphs (by the theorem of Erdős above). On the other hand the family containing odd holes and odd anti-holes is chi bounding, since the strong perfect graph theorem tells us that all graphs forbidding these as induced subgraphs are perfect - they satisfy $\chi(G) = \omega(G)$.

To prove that an hereditary class is chi-bounded, one must prove that for all k, the graphs in the class of clique number at most k have bounded chromatic number. This is often proved by induction on k, and the first case $k = 2$ is not necessarily easy. What is incredibly surprising is that we do not know of an example of an hereditary class $\mathcal{C}$ such that triangle free graphs in $\mathcal{C}$ have bounded chromatic number but the whole class is not chi-bounded. This motivates the following fascinating conjecture (due to Esperet, even though some researchers might have asked before, see [TTV17])

Conjecture 6.1.3 (*Esperet*)

If $\mathcal{C}$ *is an hereditary class such that triangle free graphs have bounded chromatic number, then* $\mathcal{C}$ *is chi-bounded.*

This is easily seen to be equivalent to :

Conjecture 6.1.4

For all k *and* n*, there exists* p *such that every graph* G *of clique number at most* k *and chromatic number at least* p *has a triangle free induced subgraph of chromatic number at least* n*.*

This is true if we do not ask the subgraph to be induced by a result of Rödl [Röd77]. It is surely very difficult as it would imply a recent and difficult result of Scott and Seymour that graphs with no odd holes are chi-bounded (since for this class triangle free graphs are simply bipartite graphs) and is open even for $n = 4$ (this is true for $n = 3$ by the previously mentioned theorem on odd holes).

The main question in this chapter is "“which hereditary classes of graphs are chi-bounded?”, or equivalently :

Problem 6.1.5

Which families of graphs $\mathcal{F}$ *are chi-bounding?*

Note that one first thing we know is that if $\mathcal{F}$ is finite, then it must contain a forest. Indeed if not, then there exists a k such that every graph in $\mathcal{F}$ contains a cycle of length less than k, and by the theorem of Erdős mentioned above, there exists graphs that have no cycle of length at most k (and therefore that do not contain any graph in $\mathcal{F}$) that have arbitrarily large chromatic number.

- The first natural question is therefore the case where $\mathcal{F}$ consists of a single forest. This is known as Gyárfás-Sumner conjecture and we will recall background on this problem in the next section.
- In section 6.3 and 6.4 we will expose ideas and results to extend Gyárfás-Sumner Conjecture to directed graphs. There are classically two ways of defining proper colourings and hence chromatic number for directed graphs : either to say that the chromatic number is just the chromatic number of the underlying undirected graph, or to say that it corresponds to a partition of the vertex set into acyclic subgraphs (instead of independent sets). In both cases, we will try to understand what substructures one must exclude to bound the chromatic number. For the first possible definition we will present results obtained in [Abo+18] with Aboulker, Bousquet, Havet, Maffray and Zamora. Then we will investigate directions for possible future research by exposing some ongoing project started with Aboulker and Naserasr on dichromatic number [ACN]
- On the other end of the spectrum, one can wonder about the case when $\mathcal{F}$ is a infinite family of cycles, and this we will discuss in a section 6.5. I will present a result obtained with Bonamy and Thomassé ([BCT14]) and expose the current status of these questions, as several nice and strong results were obtained in the past few years in this area.
- Finally we will devote a short section to a related result obtained with Penev, Thomassé and Trotignon, presented in [Cha+16], about a notion called *clique-chromatic* number. The question is now to partition the vertex set of a graph into the least possible number of parts such that no maximal clique is monochromatic (this invariant is a lower bound on classical chromatic number - e.g. any clique can be 2-coloured). In our paper we answered a question of Dufus et al [Duf+91] by proving that there exists perfect graphs of arbitrarily large clique chromatic number.

6.2 Chi-bounding Forests, the Gyárfás-Sumner Conjecture

As mentioned before, if a family $\mathcal{F}$ consists of a single graph F and $\mathcal{F}$ is chi-bounding, then F must be a forest. Asking if this is sufficient is a famous conjecture formulated independently by Gyárfás ([Gyá75]) and Sumner ([Sum81]).

Conjecture 6.2.1 (*Gyárfás-Sumner*)

If F is a forest, the class of graph excluding F as an induced subgraph is chi-bounded.

A nice argument of Gyárfás asserts that it is in fact enough to prove it for trees.

Lemma 6.2.2

If a graph H is the disjoint union of graphs H_1 and H_2. Then H is chi-bounding if and only if H_1 and H_2 are chi-bounding.

The question is still widely open, as the only trees that have been shown to be chi-bounding are :

- Stars (easy with use of Ramsey's theorem)
- Paths (Gyarfas [Gyá75; Gyá87])
- Subdivisions of stars (which generalizes the two cases above, [])
- Trees of radius 2 (Gyárfás, Szemerédi and Tuza [GST80] in the triangle-free case; Kierstead and Penrice [KP94] in the general case)
- Some special trees of radius 3 (Kierstead and Zhu [KZ04])
- Trees obtained from a tree of radius two by subdividing some edges incident with the root. Seymour and Scott : [SS18b].

Let us explain for example why stars is an easy case : indeed forbidding cliques of size k and stars $K_{1,l}$ is exactly saying that every neighbourhood in the graph has clique number at most $k-1$ and independence number at most l. By Ramsey's classical theorem this implies that the all neighbourhoods have bounded size, and by degeneracy this implies that the chromatic number is bounded.

Let us also include the proof of for paths as the idea is easy and has been reused a lot after. Here is a first lemma that illustrates the role of controlling the chromatic number of neighbourhoods.

Lemma 6.2.3

Let G be a connected graph distinct from K_1 and assume that for every vertex x the chromatic number of $G[N(x)]$ is at most t. Then for any vertex x there exists an induced path starting in x of length at least $\chi(G)/t$.

We prove this by induction on the size of G. First notice that if x is universal then $t \geq \chi(G) - 1$ and the lemma just says that no vertex is isolated. If not then amongst the connected components of $G \setminus N[x]$, there must exist one component C of chromatic number at least $\chi(G) - t$ (otherwise G can be coloured with less than $\chi(G)$ colours). Now take y to be a neighbour of x that has a neighbour in C (y exists because G is connected) and apply induction on $G[C \cup \{y\}]$. We obtain in this graph an induced path starting in y of length at least $(\chi(G) - t)/t$, and by adding x, we get the lemma.
Now with this lemma and since in (K_k, P_p)-free graphs, neighbourhoods of vertices are (K_{k-1}, P_p)-free, one can prove easily by induction on k that (K_k, P_p)-free graphs have chromatic number at most $(p-1)^{k-2}$.

One idea that is important in proof above is this idea of constructing an object progressively by keeping still a reserve of high chromatic number (here the connected component C).

Before ending this discussion, let us mention two related results. Scott proved the following very nice "topological" version of the conjecture.

Theorem 6.2.4 (*Scott, [Sco97]*)

For every tree T, the class of graphs excluding all subdivisions of T is chi-bounded

When one does not ask the tree to be induced, everything is trivial as it is folklore (see for instance [GST80]) to prove that every k-chromatic graph contains every tree on k-vertices as a subgraph. Hajnal and Rödl proved another extension of the conjecture (but apparently denied by Hajnal, see [KR96])

Theorem 6.2.5 (*Halnal and Rödl*)

For every tree T and every integer $n \geq 0$, all T-free graphs not containing the complete bipartite $K_{n,n}$ as a subgraph (not necessarily induced) have bounded chromatic number.

Now if a T-free graph has clique number at most k and has very large chromatic number it must by the theorem above contain a $K_{k,k}$ and since both sides cannot be cliques, it follows that the graph must contain an induced C_4. This therefore proves the following weakening of Gyárfás and Sumner Conjecture.

Theorem 6.2.6

For every tree T, the class of $\{T, C_4\}$-free graphs is chi-bounded.

6.3 Extending Gyarfas Sumner : Orientations

Here is a way to extend the conjecture of Gyárfás and Sumner. Assume G is a graph of large chromatic number and bounded clique number and orient its edges. What subdigraphs are forced to be present, if any? For example if T is a tree for which we know Gyárfás Sum conjecture to be true and $\overrightarrow{T}$ some orientation of it, the class of graphs that admit an orientation without any induced subdigraph isomorphic to $\overrightarrow{T}$ is clearly a super class of $\operatorname{Forb} T$, and one can wonder for which choice of $\overrightarrow{T}$ this gives a chi-bounded class.

This question was raised first by Gyárfás [Gyá90] but before getting into that, let us again note that in this oriented setting, if we do not ask the subdigraphs to be induced, the same questions can be asked but the answer is radically different. Burr [Bur80] proved that every orientation of a $(k-1)^2$-chromatic graph contains every oriented tree of order k and conjectured that replacing $(k-1)^2$ by $2k-2$ should be enough. The best known result in that directions is $(k^2/2 - k/2 + 1)$ (Addario-Berry et al. [Add+13]). This is still also open for tournaments (that is every tournament of order $2k-2$ contains every oriented tree of order k as a subdigraph) and is referred to as Sumner's conjecture, see for example [DH18] for the most recent results on the topic.

Let us go back into our world of induced subgraphs, and let us say that a set $\mathcal{F}$ of oriented trees is chi-bounding if the class of graphs that can be oriented without any induced subdigraph in $\mathcal{F}$ is chi-bounded.

Since Gyárfás knew that paths were chi-bounding in the unoriented setting, it was natural to investigate which orientations of paths would be χ-bounding. Let us review the easy cases. We will represent the paths schematically : for example $\rightarrow\leftarrow\rightarrow$ represents the oriented path on 4 vertices with alternating directions of arcs. Also we denote by $\overrightarrow{TT_n}$ the *transitive tournament* on n vertices, that is the only acyclic orientation of a complete graph on n vertices, and by $\overrightarrow{C_n}$ the cyclic orientation of a cycle on n vertices.

- $\to\to$: By a result of Ghouila-Houri [Gho62] a graph admits an orientation without $\to\to$ (these are called quasi-transitive orientation) if and only if it has a transitive orientation, that is an orientation both acyclic and quasi-transitive. In other words, these are the comparability graphs that we already encountered in other parts of this document. Note that if a graph has a transitive orientation, then cliques correspond to directed paths and according to a classical theorem, due independently to Gallai [Gal68], Hasse [Has65], Roy [Roy67], and Vitaver [Vit62], the chromatic number of a digraph is at most the number of vertices of a directed path of maximum length : this implies that comparability graphs are perfect. Hence the graphs that can be oriented without any $\to\to$ are perfect graphs, and therefore χ-bounded.
- $\to\leftarrow$: Oriented graphs avoiding this must have complete in-neighbourhood for every vertex. But then if a graph of clique number k can be oriented this way, it means the in-degree of every vertex is at most $k-1$, and this easily implies that $\chi(G)$ is at most $2k$ (by degeneracy), so again this defines a chi-bounded class.
- $\to\leftarrow\to$: Gyárfás proved that this orientation is not chi-bounding. One can consider the line digraph of the transitive tournament TT_n (the line digraph of a digraph D is the digraph whose vertices are the arcs of D and where ef is an arc if and only id the tail of e is the head of f). It is easily seen to be $\to\leftarrow\to$-free and triangle free (whatever orientation of the triangle) and that its underlying graph has not bounded chromatic number (it is $\log_2(n)$).
- $\to\to\to$: Kierstead and Trotter [KT92] proved that $\to\to\to$ is not χ-bounding by constructing $(\overrightarrow{TT_3}, \to\to\to)$-free oriented graphs with arbitrary large chromatic number (with an easy oriented variant of Zykov construction of triangle free high chromatic graphs).

In [Abo+18] we proved several additional results (we gave the exact bounds for several cases above) and conjectured that the last 4-vertex path should be chi-bounding. We supported this conjecture by proving the triangle free case (and a bit more).

Theorem 6.3.1 (*[Abo+18]*)

- *Triangle free graphs that can be oriented without induced* $\to\to\leftarrow$ *have chromatic number at most* 3
- *Graphs that can be oriented without induced* $\overrightarrow{TT_3}$ *and* $\to\to\leftarrow$ *have chromatic number at most* 4

Since every orientation of K_4 contains a $\overrightarrow{TT_3}$, the second point was a first step for the K_4-free graphs case. Eventually Chudnovsky et al. proved the validity of the conjecture in [CSS19].

After paths, we investigated in [Abo+18] the same question for oriented stars. Let us denote by $S_{k,l}$ the orientation of a star with $k+l$ leaves obtained by orienting k arcs towards the center and l out from the center. We conjectured in our paper that all oriented stars should be chi-bounding. The case when k or l is zero is easy because it is Ramsey type of argument again (like in Section 6.2). The case $k=l=1$ corresponds to the case $\to\to$ already described. We again proved the first step of the conjecture by proving the following.

Theorem 6.3.2 (*[Abo+18]*)

Let S be any oriented star.

- *Triangle free graphs that can be oriented without induced S have bounded chromatic number.*
- *Graphs that can be oriented without induced $\overrightarrow{TT_3}$ and S have bounded chromatic number.*

As for paths, the second point was a first step towards the K_4-free case of the conjecture, and as for paths, Chudnovsky et al. proved in [CSS19] that all stars were chi-bounding.

Since $\rightarrow\leftarrow\rightarrow$ and $\rightarrow\rightarrow\rightarrow$ are not chi-bounding any oriented tree that contains these could not chi-bounding, we conjecture that this is exactly the case. The remaining paths could be a next step : proving that paths with like $\rightarrow\rightarrow\leftarrow\leftarrow\rightarrow\rightarrow\leftarrow\leftarrow$ using the ideas of [CSS19] could be a attainable project for a student.

One could wonder whether Gyárfás-Sumner Conjecture could be extended by asking whether for any tree, at least one orientation of it is chi-bounding. This is not the case by considering the comb tree : a path with one pendant vertex attached to each vertex of the path. It is easy to see that if it is long enough (at least 4 vertices), any orientation of the comb contains either $\rightarrow\leftarrow\rightarrow$ or a $\rightarrow\rightarrow\rightarrow$.

Further discussions on this problem can be found in our paper [Abo+18].

6.4 Acyclic Colouring of Directed Graphs

The content of this section comes from an work in progress with Aboulker and Naserasr [ACN]. The goal is again to extend the considerations of this chapter to directed graphs (digraphs) but in a different way that what was explained in Section 6.3.

Recall that for us, digraphs can contain cycles of length 2, and if we forbid those, then the objects are referred to as oriented graphs. A *tournament* is an orientation of a complete graph. The notion of induced subgraph naturally extends to digraphs and therefore we also define hereditary classes of digraphs as classes closed under induced subdigraphs, or equivalently as defined by the exclusion of some family $\mathcal{F}$ of digraphs as induced subdigraphs. We will write $\overrightarrow{Forb}(\mathcal{F})$ in this context.

Acyclic colouring For digraphs, define a k-colouring to be a partition of the vertex set into k acyclic subdigraphs (i.e. no directed cycle is monochromatic). The *acyclic chromatic number*, or *dichromatic number* of a digraph, denoted $\vec{\chi}$, is then the minimum k for which the digraph admits a k-colouring. This notion was first introduced by Neumann-Lara [Neu82] and has received a lot of attention in the past decade, because it seems the natural way to extend the notion of chromatic number of undirected graphs as several basic classical results from graph colouring have their analogue for $\vec{\chi}$ ([Gol14; Bok+04; HM11b; HM11a; Moh03; Moh10]).

There are many open questions about this invariant. Perhaps the most natural one is to wonder about the links with the usual chromatic number, it was raised at the end of the seventies.

Conjecture 6.4.1 (*Erdős and Neumann-Lara, 1979, [Erd79]*)

For any integer p, there exists an integer $f(p)$ such that if G is a graph such that all its orientations *have dichromatic number at most p, then G has chromatic number at most $f(p)$.*

Little is known on this conjecture : it is trivial that $f(1) = 2$ but it is not even known if $f(2)$ exists. Recently Mohar and Wu proved that this is true for the fractional chromatic number instead of the chromatic number [MW16].

Another very nice conjecture due to Neumann-Lara [Neu85] is that every oriented planar graph has dichromatic number at most 2. Bokal et al. [Bok+04] proved it with 3 instead of 2 and Harutyunyan and Mohar proved it if the graph has directed girth at least 5 [HM17b].

Note that each (unoriented) graph G can be viewed as a digraph by replacing each edge with the two possible corresponding arcs. This digraph will be denoted by $\overleftrightarrow{G}$. Unoriented graphs seen as such digraphs then correspond to $\overrightarrow{Forb}(\overrightarrow{K_2})$, where $\overrightarrow{K_2}$ denotes the oriented graph consisting of one arc. Note also that we trivially have $\vec{\chi}(\overleftrightarrow{G}) = \chi(G)$ since to every edge of G corresponds a circuit of length 2.

The Gyárfás-Sumner conjecture can be restated as follows :

Conjecture 6.4.2 (*Gyarfas-Sumner; restated*)

For any forest F and integer t, the class $\overrightarrow{Forb}(\{\overrightarrow{K_2}, \overleftrightarrow{F}, \overleftrightarrow{K_t}\})$ is of bounded dichromatic number.

Pursuing the questions of this chapter, we will investigate here which hereditary classes of digraphs have bounded dichromatic number, or equivalently which families $\mathcal{F}$ of digraphs are such that $\overrightarrow{Forb}(\mathcal{F})$ has bounded dichromatic number.

The following classes of digraphs, each of unbounded dichromatic number, provide a necessary conditions for $\mathcal{F}$.

1. The class of complete symmetric digraphs (because $\vec{\chi}(\overleftrightarrow{K_n}) = n$).
2. For all fixed integer g, the class of symmetric digraphs $\overleftrightarrow{G}$ whose underlying graph has girth at least g (by Erdős Theorem on undirected graphs).
3. The class of all tournaments (replace each vertex of a directed triangle by a directed triangle and iterate).
4. For all fixed integer g, the class of oriented graphs whose underlying graph has girth at least g. This is an extension of Erdős Theorem to acyclic colouring (Theorem 2.1 in [HM12]).

Thus to have a bound on dichromatic number of a class $\overrightarrow{Forb}(\mathcal{F})$ of digraphs, $\mathcal{F}$ must contain these four types of digraphs :

- A complete symmetric digraph $\overleftrightarrow{K_k}$ for some integer k, because of item 1.
- A symmetric forest $\overleftrightarrow{F_1}$ because of item 2.
- A family $\mathcal{T}$ of tournaments such that tournaments forbidding these have bounded dichromatic number because of item 3. In the seminal paper [Ber+13], Berger et al. gave a structural characterization of the case where $\mathcal{T}$ is a singleton, these tournaments are called *heroes*.

- An oriented forest $\overrightarrow{F_2}$ because of item 4.

It follows that no chi-bounding foribidden family $\mathcal{F}$ can be of size 1, the only such family of size 2 is $\{\overrightarrow{K_2},\overleftrightarrow{K_2}\}$ which yields only empty graphs, so we looked at families of size 3. There are three cases

1. $\overrightarrow{Forb}(\{\overleftrightarrow{K_t}, \overrightarrow{K_2}, \overleftrightarrow{F}\})$ which corresponds to the case of undirected graphs and Gyárfás-Sumner.
2. $\overrightarrow{Forb}(\{\overleftrightarrow{K_k}, \overline{K}_\alpha, H\})$, where H is some hero tournament.
3. $\overrightarrow{Forb}(\{\overleftrightarrow{K_2}, \overrightarrow{F}, H\})$, where H is some hero tournament, and $\overrightarrow{F}$ some oriented forest.

We solved the second case by proving the following theorem.

Theorem 6.4.3

Let $k, \alpha \geq 1$ be integers and let H be a hero. The class $\text{Forb}\,(\overleftrightarrow{K}_k, \overline{K}_\alpha, H)$ *has bounded dichromatic number if and only if $k \leq 2$ or H is a transitive tournament*

The only if part is due to the following construction. Let $\overleftrightarrow{G}$ be the biorientation of a graph G with arbitrarily large girth and chromatic number. We fix an arbitrary enumeration $v_1, \dots, v_n$ of the vertices of $\overleftrightarrow{G}$ and create a semi-complete (i.e. $\overline{K}_2$-free) digraph D as follows: if v_iv_j with $i < j$ is a non-edge of $\overleftrightarrow{G}$, then v_iv_j is an arc of D. It is clear that this has arbitrarily large chromatic number, and belongs to $\text{Forb}\,(\overleftrightarrow{K}_3, \overline{K}_2, \overrightarrow{C_3})$. We conclude by the fact that in [Ber+13] it is proven that every hero either contains a $\overrightarrow{C_3}$ or is a transitive tournament.

The if part contains 2 statements : the $k = 2$ case was proven by Harutyunyan et al. recently in [Har+17] and the case when H is a transitive tournament is just the observation that by a classical Ramsey argument every large enough (in terms of number of vertices) digraph etiher contains a large K_k, or a large $\overline{K}_\alpha$, or a large tournament. It is enough since by an easy induction argument one can prove that every tournament on 2^t vertices contains a TT_t.

For the third case above, we propose the following conjecture (whose only if part is true and explained after).

Conjecture 6.4.4

Let H be a hero and let $\overrightarrow{F}$ be an oriented forest. Then $\text{Forb}\,(\{\overleftrightarrow{K}_2, \overrightarrow{F}, H\})$ *has bounded dichromatic number if and only if:*

- *$\overrightarrow{F}$ is the disjoint union of oriented stars,*
- *or H is a transitive tournament.*

The only if part is due to the following construction of a family of oriented graphs $D_1, \dots, D_k, \dots$ such that, for every i, $\overrightarrow{\chi}(D_i) = i$, the only oriented forest D_i contains is a union of stars, and the only tournament D_i contains is a transitive one. Take $D_1 = K_1$ and D_{i+1} is the digraph made of three disjoint copies D_i^1, D_i^2, D_i^3 of D_i plus a vertex v such that $v \rightarrow D_i^1 \rightarrow D_i^2 \rightarrow D_i^3 \rightarrow v$.

If both conditions are realized (that is, F is a disjoint union of stars AND H is a transitive tournament) then the statement is true, and even more. Indeed the result of Chudnovsky et al. mentioned in section 6.3 about oriented stars proves that the underlying graphs of oriented graphs in $\overrightarrow{Forb}(\overleftrightarrow{K}_2, S, TT_t)$ have bounded chromatic number [CSS19] for every oriented star S (and the same follows easily for a disjoin union of oriented stars).

Note also that as we explained before every large enough tournament contains some fixed transitive tournament, so when trying to decide which classes of oriented graphs have bounded dichromatic number, forbidding a transitive tournament is equivalent to forbidding some clique in the underlying graph. Proving the "if" part of the conjecture above in the case where H is a transitive tournament therefore can be thought of as : for every oriented forest $\overrightarrow{F}$ and every k, every oriented graph of large enough dichromatic either contains $\overrightarrow{F}$ or some oriented clique on k vertices as an induced subdigraph. As we have seen in the previous section, for some forests we know that this is even true for chromatic number instead of dichromatic number. The smallest oriented paths for which this is not the case are $\rightarrow\leftarrow\rightarrow$ and $\rightarrow\rightarrow\rightarrow$. In [ACN] we make a first small step by proving that oriented graph forbidding this path and a clique on 4 vertices have bounded dichromatic number.

6.5 Chi-bounding Families of Cycles

Let us now go back to undirected graphs. As we saw in the introduction, a single cycle, or any graph that contains a cycle, cannot be chi-bounding by itself. But what if we forbid several holes of different length at the same time? Again because of Erdős'sTheorem, we know that we need to forbid infinitely many. And there are of course cases that work : one of the simplest result of graph theory is the fact that if we forbid all odd cycles, then the graph is bipartite. What about other infinite families?

The first question is whether ANY infinite family is chi-bounding. A simple construction proves that this is not the case. Using again Erdős's Theorem 6.1.1 one can define recursively a sequence F_i of graphs such that $\chi(F_i) \geq i$ and $\mathrm{girth}(F_i) > |F_{i-1}|$. Now if we denote by $\mathcal{F}$ the set of holes that do not occur in any F_i, then $\mathcal{F}$ is not chi-bounding and is infinite (since it contains at least all holes of length $|F_i|$, $i \geq 4$).

So to get a χ-bounded class one must forbid an infinite family of holes whose length have some constraints.

In [BCT14], with Bonamy and Thomassé, we proved one of the first non trivial result of this kind by answering a question of Kalai and Meschulam (in [KM]) regarding cycles of length 0 mod 3.

Theorem 6.5.1 (*Bonamy,C.,Thomassé,[BCT14]*)

There exists k such that any graph of chromatic number greater than k contains an induced cycle of length 0 mod 3.

It was not an easy theorem to prove (see later for short hints about the proof) but note that it it is just the triangle free case. We were not able to prove that the family of holes (meaning induced cycle of length at least 4) of length 0 mod 3 is chi-bounding, we just settled the triangle-free case. Moreover it might be that the real answer in the theorem above is $k = 3$.

Conjecture 6.5.2

Graphs with no induced cycle of length 0 mod 3 are 3 colourable.

On our theorem above, we did not try to get the best k possible, but it would be in any case very high, so certainly our approach could not be of use to tackle this question.

The same question without the word "induced" (this defines of course a much smaller class) is true, let us explain the reason why. We first remove any edge e (and stay in the class since we are not talking about induced subgraph anymore) and apply induction to get a valid 3-colouring. If the endpoints x and y of e are coloured differently we win, so we can assume they belong to the same colour class, w.l.o.g colour 1. Let now D be the the oriented graph obtained from $G \backslash e$ orienting every edge from colour i to $i+1$ (mod 3), and define X to be the set of vertices of D that are reachable from x by a directed path. If y does not belong to X then it is easy to see that by cyclically shifting the colour of every vertex in X (that is give colour $i+1 \bmod 3$ to a vertex coloured i), the colouring we get is still a good colouring and since x is now of colour 2, we get a proper colouring for G. So it means that y is in X and similarly x is in Y if we define Y symmetrically starting with y. By concatenating those two paths we get a directed closed walk with vertices coloured 1,2,3,1,2,3,... A minimal such object in the graph necessarily induces an (induced) cycle of $G \setminus e$ of length $0 \bmod 3$, which is a contradiction. As pointed out, we get in the end of this argument an induced cycle of length $0 \bmod 3$. So this proof would work for conjecture 6.5.2 as long as we could initiate the induction, that is if there always exists an edge whose removal yields a graph that has still no induced cycle of length $0 \bmod 3$. This was for some time a conjecture, until Wrochna ([Wro18]) found a counterexample (see Figure 6.1)

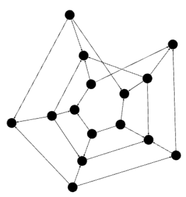


Figure 6.1: A Graph with no $0 \bmod 3$ induced cycle but such that each edge removal produces one

Since out proof of Theorem 6.5.1, a impressive series of papers on colouring hereditary classes ([SS16; CSS16a; CSS17a; SS15; SS17a; SS17b; CSS16b; SS17c; SS17d; Chu+17b; SS17e; CSS19; CSS17b; SS18b]), authored by Scott, Seymour (and sometimes Chudnovsky) established several beautiful results, extending our theorem to much more cases. Let us mention that as these types of questions were first asked by Gyárfás in [Gyá87]. At that time, the strong perfect graph theorem was still a conjecture, and in fact it was even unknown whether Berge graphs (recall that these are graphs with no odd hole and no odd antihole) were chi-bounded (in fact, to my knowledge, the first proof of this came only with the proof of the strong perfect graph theorem, whose result is much stronger than just chi-boundedness). Gyarfas made then three conjectures (the third one implying the first two) about odd holes, long holes, and long odd holes.

Conjecture 6.5.3 (*Gyárfás*)

- *The family of odd-holes (length at least 5) is chi-bounding*

- *For any l, the family of holes of length at least l is chi-bounding.*
- *For any l, the family of odd holes of length at least l is chi-bounding.*

For odd holes, there has been several attempts to prove it that were not successful. Hoang and Mc Diarmid made a very nice conjecture that, if true, easily implies a bound $\chi \leq 2^{\omega}$.

Conjecture 6.5.4 (*Hoang,McDiarmid*)

Every odd-hole free graph admits a partition of its vertex set such that no maximum clique is contained in one of the parts.

On the other hand, the case of even-hole free graphs is non trivial and follows from a nice and not easy structural result by Addario-Berry, Chudnovsky, Havet, Reed and Seymour ([Add+08]) who proved that every such graph always contain a vertex that is bisimplicial - a vertex whose neighbourhood can be partitioned into two cliques. This implies that even hole free graphs satisfy $\chi \leq 2\omega$.

All of the conjectures of Gyárfás written above are now theorems. Scott and Seymour proved them one after another, and eventually proved in [SS17e] the following result that contains them all :

Theorem 6.5.5 (*Scott, Seymour, [SS17e]*)

For any integers n and p_i, q_i, for $1 \leq i \leq n$, let C be the class of graphs that do not contain n pairwise anticomplete holes $H_1, ..., H_n$ where H_i has length p_i modulo q_i for $1 \leq i \leq n$. Then C is chi-bounded.

For $n = 1$, the theorem above contains in particular that for any p and q, the set of integers equal to $p \bmod q$ is chi-bounding (we say by abuse of language that a set of integers X is chi-bounding if the set of holes with length in X is chi-bounding). This is a special case of the following open question.

Conjecture 6.5.6 (*Scott, Seymour,[SS18a]*)

Every set of integers with bounded gaps (its complement does not contain arbitrarily long list of consecutive integers) is chi-bounding.

This can be strengthened even more.

Conjecture 6.5.7 (*Scott, Seymour,[SS18a]*)

For any integer q, the class of graphs without holes of q consecutive length is chi-bounded.

Seymour and Scott proved only the triangle-free case in 3.1. How to relax that even more? The lower (resp.upper) density of a set of integers is the $\liminf$ (resp. $\limsup$) of the sequence $|X \cap [1; n]|/n$. A set of integers of bounded gaps has clearly non zero lower density. Could it be sufficient?

Conjecture 6.5.8 (*Scott, Seymour,[SS18a]*)

Every set of integers with strictly positive lower density is chi-bounding.

In fact Seymour and Scott (see [SS18a]) even conjecture that this condition is necessary. Note that one cannot replace lower by upper density : using Erdős Theorem (like we did to prove that some infinite families are not chi-bounding at the beginning of this section), it is possible to construct families with strictly positive upper density that are not chi-bounding.

A lot of other nice related conjectures can be found into the beautiful survey on χ-boundedness [SS18a] by Seymour and Scott.

Links with topological aspects

As mentioned above, Theorem 6.5.1 about graphs with no cycles of length 0 mod 3 was the answer to a conjecture of Kalai and Meschulam, which did not come from Gyárfás's questions, but from more topological considerations. The independence complex $\mathcal{I}(G))$ of a graph G is the simplicial complex formed by independent sets of G. Some very nice combinatorial results were obtained by studying topological properties of this complex. For example bounds on the connectivity (the dimension of the smallest hole in $\mathcal{I}(G))$) can be used to prove the existence of linear independent sets (see Aharoni and Haxell [AH00]). But can we go beyond large independent sets and prove small chromatic number? Is it true that if the independent complex is 'simple' for G and all of its subgraphs, then G has bounded chromatic number.

The idea was developed by Kalai and Meschulam, and the parameter they proposed is to consider, for a given graph H, the sum $bn(H)$ of all reduced Betti numbers of $\mathcal{I}(H)$ (i.e. the sum of the number of independent holes in each dimension, or more precisely the sum of the ranks of all homology groups). They conjectured that if G has large chromatic number, then one of its induced subgraphs H has large $bn(H)$. Observe that large cliques have in particular large parameter bn, and that this conjecture would imply the existence of a "complex" induced subgraph (at least with respect to some parameter which is typically large for complete graphs). Now one can notice that $\mathcal{I}(C_6)$ has two non-equivalent (1-dimensional) holes, while $\mathcal{I}(C_4)$ and $\mathcal{I}(C_5)$ first non trivial homology groups have rank 1. This remark generalizes as follows: the (unique) non trivial homology group of $\mathcal{I}(C_n)$ has rank 1 if n has length 1 or 2 $\bmod 3$, otherwise it has rank 2. Therefore, a graph only inducing graphs H with $bn(H) \leq 1$ does not have induced cycles of length $0 \bmod 3$. Hence, if one wants to show the first nontrivial case of the Kalai-Meshulam conjecture, i.e. that large chromatic number implies the existence of an induced subgraph H with $bn(H) > 1$, it would suffice in particular to show that every graph with large chromatic number has an induced $3k$-cycle, and that is precisely what we proved in [BCT14].

Scott and Seymour noted that one can push the above argument : a graph only inducing graphs H with $bn(H) \leq k$ cannot contain k induced cycles of length $0 \bmod 3$ that are all pairwise anticomplete. Combining this fact with their Theorem 6.5.5, they obtained as a corollary ([SS17e]) a proof of Kalai and Meschulam conjecture stated above.

About the proof of Theorem [BCT14]

The proof is quite technical so we will just here give hints about it. By analogy with parity, we call *trinity* of an integer its residue mod 3 and *trinity graph* a graph with no circuit whose length is of trinity 0.

The first ingredient of several of the proofs in this area is to consider a fixed vertex and then partition the graph into layers depending on the distance to this root vertex. Note that if a graph has chromatic number at least k, then there must be a layer of chromatic number at least $k/2$.

A second ingredient is what we call a *trinity changing path system* (TCPS). It is a sequence of graphs such that we can go through each graph with two induced paths with different trinities.

Formally, a *brick B with extremities* (x, y) is a connected graph that admits two induced xy-paths with different trinities. In particular $x \neq y$ and xy is not an edge in B. A TCPS of *order* k is obtained by considering a sequence of pairwise disjoint and non-adjacent bricks $B_1, \ldots, B_k$ with extremities $(x_1, y_1), \ldots, (x_k, y_k)$ (respectively), then identifying y_i with x_{i+1} for $i \in \{1, \ldots, k-1\}$. The vertex x_1 is called the *origin* of the TCPS. We were able to prove (with ideas similar to the ideas of Gyárfás's proof of the existence of long induced path explained in Section 6.2) that if a connected trinity graph G has sufficiently large chromatic number then every vertex of G is the origin of a large order TCPS.

Now let us give hints on how we can use these structures. Consider at a layer L_i of large chromatic number, and C a connected component of L_i of large chromatic number. Let u be a vertex of L_{i-1} with a neighbour x in C. There must be in $C \setminus N(u)$ a component C' or large chromatic number (recall that the graph is triangle free, so $N(u)$ is a stable set), so let $x \in C$ be a neighbour of u with a neighbour in C' and find a large order TCPS of origin x in $C' \cup \{x\}$. Observe that if some vertex v of $N_{\ell-1}$ has a neighbour y in some block B_i of T, then we can close a cycle using an path between u and v using only layers above L_{i-1} and some induced uy-path P on the TCPS. Since P has two trinity choices when traversing each brick, the only way to avoid a trinity 0 cycle is that v itself sees many bricks. Precisely, v must see every pair of bricks $B_j \cup B_{j+1}$ where $1 \leq j < i$. In particular, if T has $2k$ bricks for some large value k, the set X of vertices in $N_{\ell-1}$ which see some B_i with $k \leq i \leq 2k$ is such that between every pair of vertices in X there are many independent induced paths. If additionally these two vertices are not joined by an edge, then at most one of these path is of trinity 0 and all the others have the same trinity, either 1 or 2 otherwise we would get a trinity 0 induced cycle. If we have a large independent sets in X this is a good start for our purposes. To use these considerations, our goal is to find inside a layer a large chromatic set that is dominated in a layer above by a independent set. This is what we call shadow. This is not always possible to have one but we were able to prove that in a trinity graph of large chromatic number, either there is a shadow, or there is what we call an antishadow, which is (roughly, there are additional conditions to serve our purposes) a large chromatic set that is dominated in a layer *below* by a independent set.

Using these tools the proof goes into several steps by finding a special graph H that splits the difficulty of the problem, that is knowing that H is forbidden or knowing that H is a subgraph both lower the difficulty of the problem. The graphs mentioned below are represented on Figure 6.2. The steps for the proof are as follows (even though the first step is in fact the harder and the one we prove last).

- C_5-free trinity graph have bounded chromatic number
- Extended C_5-free trinity graphs have bounded chromatic number
- Doubly Extended C_5-free trinity graphs have bounded chromatic number
- Trinity graphs have bounded chromatic number

6.6 Clique Colouring

In this last section we want to present a result obtained in [Cha+16] on a related topic. It deals with perfect graphs, and concerns the notion of clique colouring. A *clique-colouring* of a graph

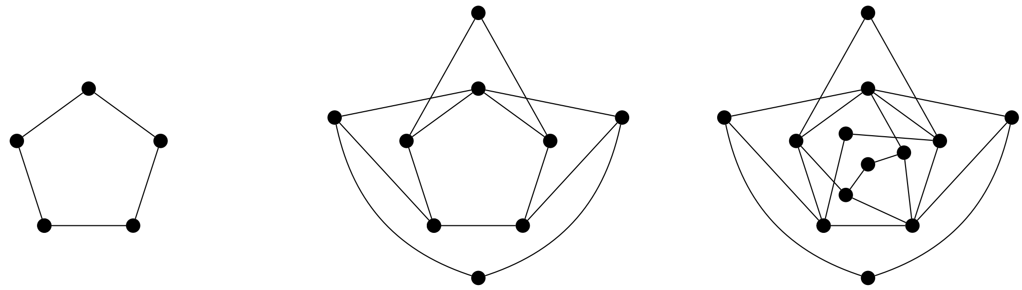


Figure 6.2: A C_5, an extended C_5 and a doubly extended C_5.

G is an assignment of colours to the vertices of G in such a way that no inclusion-wise maximal clique of size at least two of G is monochromatic (as usual, a set of vertices is *monochromatic* if all vertices in the set received the same colour). A *k-clique-colouring* of G is a clique-colouring $\varphi : V(G) \to \{1, \dots, k\}$ of G. G is *k-clique-colourable* if there exists a k-clique-colouring of G. The *clique-chromatic number* of G, denoted by $\chi_C(G)$, is the smallest integer k such that G is k-clique-colourable. Note that every proper colouring of G is also a clique-colouring of G, and so $\chi_C(G) \leq \chi(G)$. Furthermore, if G is triangle-free, then $\chi_C(G) = \chi(G)$ (since there are triangle-free graphs of arbitrarily large chromatic number, this implies that there are triangle-free graphs of arbitrarily large clique-chromatic number). However, if G contains triangles, $\chi_C(G)$ may be much smaller than $\chi(G)$. For instance, if G contains a dominating vertex, then $\chi_C(G) \leq 2$ (we assign the colour 1 to the dominating vertex and the colour 2 to all other vertices of G), while $\chi(G)$ may be arbitrarily large. Note that this implies that the clique-chromatic number is not monotone with respect to induced subgraphs, that is, there exist graphs H and G such that H is an induced subgraph of G, but $\chi_C(H) > \chi_C(G)$. (In particular, the restriction of a clique-colouring of G to an induced subgraph H of G need not be a clique-colouring of H.)

It was shown in [GHM03] that for any graph H, the class of graphs that do not contain H as an induced subgraph has a bounded clique-chromatic number if and only if all components of H are paths. The *clique number* of a graph G, denoted by $\omega(G)$, is the maximum size of a clique of G. A graph G is *perfect* if all its induced subgraphs H satisfy $\chi(H) = \omega(H)$. It was asked in [Duf+91] whether perfect graphs have a bounded clique-chromatic number. It has since been shown that graphs from many subclasses of the class of perfect graphs are 2- or 3-clique-colourable [AST91; Bac+04; CL17; Déf06; Duf+91; MS99; Pen16]. There are well-known examples of perfect graphs of clique-chromatic number three (one example is the graph obtained from the cycle of length nine by choosing three evenly spaced vertices and adding edges between them), but until now, it was not known whether there were any perfect graphs of clique-chromatic number greater than three. The main result of [Cha+16] is the following theorem.

Theorem 6.6.1

There exist perfect graphs of arbitrarily large clique-chromatic number.

Thus, the question from [Duf+91] mentioned above has a negative answer. We proved Theorem 6.6.1 by exhibiting, for each integer $k \geq 2$, a perfect graph G_k of clique-chromatic number

$k + 1$. The graph G_k is obtained from cobipartite graphs (i.e. complements of bipartite graphs) by repeatedly applying the operation of gluing graphs along a clique. The fact that G_k is perfect follows from the fact that cobipartite graphs are perfect, together with the fact that the operation of gluing along a clique preserves perfection (that is, if two perfect graphs are glued along a clique, then the resulting graph is also perfect). Note also that it is immediate from the construction that G_k does not contain any induced cycle of length at least five; furthermore, G_k does not contain the complement of any odd cycle of length at least five as an induced subgraph.

It was asked in [Pen16] whether, if c is a positive integer and $\mathcal{G}$ is a hereditary class such that every graph in $\mathcal{G}$ is either c-clique-colourable or admits a clique-cutset, there must exist a positive integer d such that every graph in $\mathcal{G}$ is d-clique-colourable. Our construction of the family $\{G_k\}_{k=2}^{\infty}$ implies that this question has a negative answer (even if we restrict our attention to the case when all graphs in the class $\mathcal{G}$ are perfect). Indeed, let $\mathcal{G}$ be the class of all induced subgraphs of the graphs G_k (with $k \geq 2$). Then $\mathcal{G}$ is a hereditary class (each of whose members is a perfect graph), and every graph in $\mathcal{G}$ is either cobipartite (and therefore 2-clique-colourable [Pen16]) or admits a clique-cutset. However, $\mathcal{G}$ contains graphs of arbitrarily large clique-chromatic number (because $G_k \in \mathcal{G}$ and $\chi_C(G_k) = k + 1$ for each $k \geq 2$).

Bibliography

[Abo+18] Pierre Aboulker, Jørgen Bang-Jensen, Nicolas Bousquet, Pierre Charbit, Frédéric Havet, Frédéric Maffray, and Jose Zamora. "χ-bounded families of oriented graphs". In: *Journal of Graph Theory* 89.3 (2018), pp. 304–326.

[ACN] Pierre Aboulker, Pierre Charbit, and Reza Naserasr. "Digraph Coloring and extension of Gyarfas-Sumner conjecture". in preparation.

[Abo+15] Pierre Aboulker, Pierre Charbit, Nicolas Trotignon, and Kristina Vušković. "Vertex elimination orderings for hereditary graph classes". In: *Discrete Mathematics* 338.5 (2015), pp. 825–834.

[Abo+12] Pierre Aboulker, Marko Radovanovic, Nicolas Trotignon, and Kristina Vušković. "Graphs that do not contain a cycle with a node that has at least two neighbors on it". In: *SIAM Journal on Discrete Mathematics* 26.4 (2012), pp. 1510–1531.

[Add+08] Louigi Addario-Berry, Maria Chudnovsky, Frédéric Havet, Bruce Reed, and Paul Seymour. "Bisimplicial vertices in even-hole-free graphs". In: *J. Combin. Theory Ser. B* 98.6 (2008), pp. 1119–1164.

[Add+13] Louigi Addario-Berry, Frédéric Havet, Cláudia Linhares Sales, Bruce Reed, and Stéphan Thomassé. "Oriented trees in digraphs". In: *Discrete Mathematics* 313.8 (2013), pp. 967–974.

[AC05] Peyman Afshani and Timothy M. Chan. "Approximation Algorithms for Maximum Cliques in 3D Unit-Disk Graphs". In: *Proceedings of the 17th Canadian Conference on Computational Geometry, CCCG'05, University of Windsor, Ontario, Canada, August 10-12, 2005*. 2005, pp. 19–22.

[AH08] Peyman Afshani and Hamed Hatami. "Approximation and inapproximability results for maximum clique of disc graphs in high dimensions". In: *Inf. Process. Lett.* 105.3 (2008), pp. 83–87.

[AKS98] Pankaj K. Agarwal, Marc van Kreveld, and Subhash Suri. "Label placement by maximum independent set in rectangles". In: *Computational Geometry* 11.3 (1998), pp. 209–218.

[AH00] Ron Aharoni and Penny Haxell. "Hall's theorem for hypergraphs". In: *Journal of Graph Theory* 35.2 (2000), pp. 83–88.

[AF04] Jochen Alber and Jirí Fiala. "Geometric separation and exact solutions for the parameterized independent set problem on disk graphs". In: *J. Algorithms* 52.2 (2004), pp. 134–151.

[Ale82] V. Alekseev. "The Effect of Local Constraints on the Complexity of Determination of the Graph Independence Number". In: *Combinatorial-Algebraic Methods in Applied Mathematics* (1982). in Russian, pp. 3–13.

[AW05] Christoph Ambühl and Uli Wagner. "The Clique Problem in Intersection Graphs of Ellipses and Triangles". In: *Theory Comput. Syst.* 38.3 (2005), pp. 279–292.

[AST91] Thomas Andreae, Martin Schughart, and Zsolt Tuza. "Clique-transversal sets of line graphs and complements of line graphs". In: *Discrete Mathematics* 88.1 (1991), pp. 11–20.

[And70] Evgeniy M Andreev. "On convex polyhedra in Lobachevskii spaces". In: *Matematicheskii Sbornik* 123.3 (1970), pp. 445–478.

[Arr+00] Richard Arratia, Béla Bollobás, Don Coppersmith, and Gregory B. Sorkin. "Euler circuits and DNA sequencing by hybridization". In: *Discrete Applied Mathematics* 104.1-3 (2000), pp. 63–96.

[AWZ17] Stephan Artmann, Robert Weismantel, and Rico Zenklusen. "A strongly polynomial algorithm for bimodular integer linear programming". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2017, pp. 1206–1219.

[AZ18] Aistis Atminas and Viktor Zamaraev. "On forbidden induced subgraphs for unit disk graphs". In: *Discrete & Computational Geometry* (2018), pp. 1–41.

[APS00] Mordecai Avriel, Michal Penn, and Naomi Shpirer. "Container ship stowage problem: complexity and connection to the coloring of circle graphs". In: *Discrete Applied Mathematics* 103.1 (2000), pp. 271–279.

[Bac+18] G. Bacsó, D. Lokshtanov, D. Marx, M. Pilipczuk, Zsolt Tuza, and E. Jan van Leeuwen. "Subexponential-time Algorithms for Maximum Independent Set in P_t-free and Broom-free Graphs". In: *CoRR* abs/1804.04077 (2018).

[Bac+04] Gábor Bacsó, Sylvain Gravier, András Gyárfás, Myriam Preissmann, and András Sebo. "Coloring the maximal cliques of graphs". In: *SIAM Journal on Discrete Mathematics* 17.3 (2004), pp. 361–376.

[Ban+06] Jørgen Bang-Jensen, Bruce Reed, Mathias Schacht, Robert Šámal, Bjarne Toft, and Uli Wagner. "On six problems posed by Jarik Nešetřil". In: *Topics in Discrete Mathematics* (2006), pp. 613–627.

[Bei+18] Jesse Beisegel, Carolin Denkert, Ekkehard Köhler, Matjaž Krnc, Nevena Pivač, Robert Scheffler, and Martin Strehler. "On the End-Vertex Problem of Graph Searches". In: *arXiv preprint arXiv:1810.12253* (2018).

[Ber61] Claude Berge. "Farbung von Graphen, deren samtliche bzw. deren ungerade Kreise starr sind". In: *Wissenschaftliche Zeitschrift* (1961).

[Ber+13] Eli Berger, Krzysztof Choromanski, Maria Chudnovsky, Jacob Fox, Martin Loebl, Alex Scott, Paul Seymour, and Stéphan Thomassé. “Tournaments and colouring”. In: *Journal of Combinatorial Theory, Series B* 103.1 (2013), pp. 1–20.

[Ber+10] Anne Berry, Jean RS Blair, Jean-Paul Bordat, and Genevieve Simonet. “Graph extremities defined by search algorithms”. In: *Algorithms* 3.2 (2010), pp. 100–124.

[BB98] Anne Berry and Jean-Paul Bordat. “Separability generalizes Dirac’s theorem”. In: *Discrete Applied Mathematics* 84.1-3 (1998), pp. 43–53.

[Bie91] Dan Bienstock. “On the complexity of testing for odd holes and induced odd paths”. In: *Discrete Mathematics* 90.1 (1991), pp. 85–92.

[Bir+17] Csaba Biró, Édouard Bonnet, Dániel Marx, Tillmann Miltzow, and Paweł Rzążewski. “Fine-Grained Complexity of Coloring Unit Disks and Balls”. In: *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*. 2017, 18:1–18:16.

[Boc+14] Adrian Bock, Yuri Faenza, Carsten Moldenhauer, and Andres J. Ruiz-Vargas. “Solving the Stable Set Problem in Terms of the Odd Cycle Packing Number”. In: *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*. 2014, pp. 187–198.

[Bok+04] Drago Bokal, Gasper Fijavz, Martin Juvan, P Mark Kayll, and Bojan Mohar. “The circular chromatic number of a digraph”. In: *Journal of Graph Theory* 46.3 (2004), pp. 227–240.

[Bon+18a] Marthe Bonamy, Edouard Bonnet, Nicolas Bousquet, Pierre Charbit, and Stéphan Thomassé. “EPTAS for Max Clique on Disks and Unit Balls”. In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. 2018, pp. 568–579.

[BCT14] Marthe Bonamy, Pierre Charbit, and Stéphan Thomassé. “Graphs with large chromatic number induce 3k-cycles”. In: *CoRR* abs/1408.2172 (2014).

[BRV] Valerio Boncompagni, Marko Radovanović, and Kristina Vušković. “The structure of (theta, pyramid, 1-wheel, 3-wheel)-free graphs”. In: *Journal of Graph Theory* ().

[Bon+18b] Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Stéphan Thomassé, and Rémi Watrigant. “Parameterized Complexity of Independent Set in H-Free Graphs”. In: 2018.

[Bon+18c] *QPTAS and Subexponential Algorithm for Maximum Clique on Disk*. 2018.

[Bou94] André Bouchet. “Circle Graph Obstructions”. In: *Journal of Combinatorial Theory, Series B* 60.1 (1994), pp. 107–144.

[BLS99] A. Brandstädt, V.B. Le, and Jeremy P Spinrad. *Graph Classes: A Survey*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 1999.

[BDN97] Andreas Brandstädt, Feodor F Dragan, and Falk Nicolai. “LexBFS-orderings and powers of chordal graphs”. In: *Discrete Mathematics* 171.1-3 (1997), pp. 27–42.

[BH08] Andreas Brandstädt and Chính T. Hoàng. "Maximum Induced Matchings for Chordal Graphs in Linear Time". In: *Algorithmica* 52.4 (2008), pp. 440–447.

[Bre+03] Anna Bretscher, Derek Corneil, Michel Habib, and Christophe Paul. "A Simple Linear Time LexBFS Cograph Recognition Algorithm". In: *Graph-Theoretic Concepts in Computer Science*. Ed. by Hans L. Bodlaender. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 119–130.

[BK98] Heinz Breu and David G. Kirkpatrick. "Unit disk graph recognition is NP-hard". In: *Comput. Geom.* 9.1-2 (1998), pp. 3–24.

[Bur80] Stefan A Burr. "Subtrees of directed graphs and hypergraphs". In: *Proceedings of the Eleventh Southeastern Conference on Combinatorics, Graph Theory and Computing, Boca Raton, Congr. Numer.* Vol. 28. 1980, pp. 227–239.

[Cab15a] Sergio Cabello. *Maximum clique for disks of two sizes*. Open problems from Geometric Intersection Graphs: Problems and Directions CG Week Workshop, Eindhoven, June 25, 2015 (`http://cgweek15.tcs.uj.edu.pl/problems.pdf`). [Online; accessed 07-December-2017]. 2015.

[Cab15b] Sergio Cabello. "Open problems presented at the Algorithmic Graph Theory on the Adriatic Coast workshop, Koper, Slovenia (`https://conferences.matheo.si/event/6/picture/35.pdf`)". In: June 2015.

[CGO10] Jérémie Chalopin, Daniel Gonçalves, and Pascal Ochem. "Planar Graphs Have 1-string Representations". In: *Discrete & Computational Geometry* 43.3 (Apr. 2010), pp. 626–647.

[Cha03] Timothy M. Chan. "Polynomial-time approximation schemes for packing and piercing fat objects". In: *J. Algorithms* 46.2 (2003), pp. 178–189.

[Cha+08] P. Charbit, M. Habib, V. Limouzy, F. de Montgolfier, M. Raffinot, and M. Rao. "A note on computing set overlap classes". In: *Information Processing Letters* 108.4 (2008), pp. 186–191.

[CDR12] Pierre Charbit, Fabien De Montgolfier, and Mathieu Raffinot. "Linear time split decomposition revisited". In: *SIAM Journal on Discrete Mathematics* 26.2 (2012), pp. 499–514.

[CHM14] Pierre Charbit, Michel Habib, and Antoine Mamcarz. "Influence of the tie-break rule on the end-vertex problem". In: *Discrete Mathematics & Theoretical Computer Science* 16.2 (2014), pp. 57–72.

[Cha+17] Pierre Charbit, Michel Habib, Lalla Mouatadid, and Reza Naserasr. "A New Graph Parameter to Measure Linearity". In: *Combinatorial Optimization and Applications - 11th International Conference, COCOA 2017, Shanghai, China, December 16-18, 2017, Proceedings, Part II*. 2017, pp. 154–168.

[Cha+12] Pierre Charbit, Michel Habib, Nicolas Trotignon, and Kristina Vušković. "Detecting 2-joins faster". In: *Journal of discrete algorithms* 17 (2012), pp. 60–66.

[CMR09] Pierre Charbit, Fabien de Montgolfier, and Mathieu Raffinot. *A Simple Linear Time Split Decomposition Algorithm of Undirected Graphs*. Tech. rep. 2009.

[Cha+16] Pierre Charbit, Irena Penev, Stéphan Thomassé, and Nicolas Trotignon. "Perfect graphs of arbitrarily large clique-chromatic number". In: *Journal of Combinatorial Theory, Series B* 116 (2016), pp. 456–464.

[CHM81] Michel Chein, Michel Habib, and Marie-Catherine Maurer. "Partitive hypergraphs". In: *Discrete Mathematics* 37.1 (1981), pp. 35–50.

[Chu+17a] M. Chudnovsky, A. Lagoutte, P. Seymour, and S. Spirkl. "Colouring perfect graphs with bounded clique number". English. In: *Journal of Combinatorial Theory.Series B* 122 (2017), pp. 757–775.

[Chu06] Maria Chudnovsky. "Berge trigraphs". In: *Journal of Graph Theory* 53.1 (2006), pp. 1–55.

[CL17] Maria Chudnovsky and Irene Lo. "Decomposing and Clique-Coloring Diamond, Odd-Hole-Free Graphs". In: *J. Graph Theory* 86.1 (2017), pp. 5–41.

[Chu+18] Maria Chudnovsky, Irene Lo, Frédéric Maffray, Nicolas Trotignon, and Kristina Vušković. "Coloring square-free Berge graphs". In: *Journal of Combinatorial Theory, Series B* (2018).

[Chu+06] Maria Chudnovsky, Neil Robertson, P.D. Seymour, and Robin Thomas. "The Strong Perfect Graph Theorem". In: *Annals of Mathematics* 164.1 (2006), pp. 51–229.

[CSS16a] Maria Chudnovsky, Alex Scott, and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. II. Three steps towards Gyárfás' conjectures". In: *Journal of Combinatorial Theory, Series B* 118 (2016), pp. 109–128.

[CSS17a] Maria Chudnovsky, Alex Scott, and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. III. Long holes". In: *Combinatorica* 37.6 (2017), pp. 1057–1072.

[CSS16b] Maria Chudnovsky, Alex Scott, and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. V. Chandeliers and strings". In: *arXiv preprint arXiv:1609.00314* (2016).

[CSS19] Maria Chudnovsky, Alex Scott, and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. XI. Orientations". In: *European Journal of Combinatorics* 76 (2019), pp. 53–61.

[CSS17b] Maria Chudnovsky, Alex Scott, and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. XII. Distant stars". In: *arXiv preprint arXiv:1711.08612* (2017).

[Chu+17b] Maria Chudnovsky, Alex Scott, Paul Seymour, and Sophie Spirkl. "Induced subgraphs of graphs with large chromatic number. VIII. Long odd holes". In: *arXiv preprint arXiv:1701.07217* (2017).

[CS05a] Maria Chudnovsky and P.D. Seymour. "The structure of claw-free graphs". In: *Surveys in combinatorics 2005*. Vol. 327. London Math. Soc. Lecture Note Ser. Cambridge: Cambridge Univ. Press, 2005, pp. 153–171.

[CS08a] Maria Chudnovsky and Paul Seymour. "Claw-free graphs. IV. Decomposition theorem". In: *Journal of Combinatorial Theory, Series B* 98.5 (2008), pp. 839–938.

[CS08b] Maria Chudnovsky and Paul Seymour. “Claw-free graphs. V. Global structure”. In: *Journal of Combinatorial Theory, Series B* 98.6 (2008), pp. 1373–1410.

[CS05b] Maria Chudnovsky and Paul D Seymour. “The structure of claw-free graphs.” In: *Surveys in combinatorics* 327 (2005), pp. 153–171.

[Chu+15] Maria Chudnovsky, Nicolas Trotignon, Théophile Trunck, and Kristina Vušković. “Coloring perfect graphs with no balanced skew-partitions”. In: *Journal of Combinatorial Theory, Series B* 115 (2015), pp. 26–65.

[Chu+05a] Maria Chudnovsky*, Gérard Cornuéjols**, Xinming Liu†, Paul Seymour†, and Kristina Vušković. “Recognizing Berge Graphs”. In: *Combinatorica* 25.2 (Mar. 2005), pp. 143–186.

[Chu+05b] Maria Chudnovsky*, Gérard Cornuéjols**, Xinming Liu, Paul Seymour, and Kristina Vušković. “Recognizing berge graphs”. In: *Combinatorica* 25.2 (2005), pp. 143–186.

[CS99] S. Cicerone and G. Di Stefano. “One the extension of bipartite to parity graphs”. In: *Discrete Applied Mathematics* 95 (1999), pp. 181–195.

[CCJ90] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. “Unit disk graphs”. In: *Discrete Mathematics* 86.1-3 (1990), pp. 165–177.

[Con+01a] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković. “Balanced 0,$\pm$1 matrices I. decomposition”. In: *Journal of Combinatorial Theory, Series B* 81.2 (2001), pp. 243–274.

[Con+01b] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković. “Balanced 0,$\pm$1 matrices II. recognition algorithm”. In: *Journal of Combinatorial Theory, Series B* 81.2 (2001), pp. 275–306.

[Con+99] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković. “Even and odd holes in cap-free graphs”. In: *Journal of Graph Theory* 30.4 (1999), pp. 289–308.

[Con+02a] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković. “Even-hole-free graphs part I: Decomposition theorem”. In: *Journal of Graph Theory* 39.1 (2002), pp. 6–49.

[Con+02b] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković. “Even-hole-free graphs part II: Recognition algorithm”. In: *Journal of Graph Theory* 40.4 (2002), pp. 238–266.

[Con+97] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković. “Universally signable graphs”. In: *Combinatorica* 17.1 (1997), pp. 67–77.

[CCR99] Michele Conforti, Gérard Cornuéjols, and MR Rao. “Decomposition of balanced matrices”. In: *Journal of Combinatorial Theory, Series B* 77.2 (1999), pp. 292–406.

[CCV04a] Michele Conforti, Gérard Cornuéjols, and Kristina Vušković. “Decomposition of odd-hole-free graphs by double star cutsets and 2-joins”. In: *Discrete applied mathematics* 141.1-3 (2004), pp. 41–91.

[CCV04b] Michele Conforti, Gérard Cornuéjols, and Kristina Vušković. “Square-free perfect graphs”. In: *Journal of Combinatorial Theory, Series B* 90.2 (2004), pp. 257–307.

[CDH13] D. Corneil, B. Dalton, and M. Habib. “LDFS-Based Certifying Algorithm for the Minimum Path Cover Problem on Cocomparability Graphs”. In: *SIAM Journal on Computing* 42.3 (2013), pp. 792–807.

[Cor+16] D. Corneil, J. Dusart, M. Habib, and E. Köhler. “On the Power of Graph Searching for Cocomparability Graphs”. In: *SIAM Journal on Discrete Mathematics* 30.1 (2016), pp. 569–591.

[CLB81] D.G. Corneil, H. Lerchs, and L.Stewart Burlingham. “Complement reducible graphs”. In: *Discrete Applied Mathematics* 3.3 (1981), pp. 163–174.

[Cor04] Derek G. Corneil. “A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs”. In: *Discrete Applied Mathematics* 138.3 (2004), pp. 371–379.

[COS09] Derek G. Corneil, Stephan Olariu, and Lorna Stewart. “The LBFS Structure and Recognition of Interval Graphs”. In: *SIAM J. Discrete Math.* 23.4 (2009), pp. 1905–1953.

[CS15] Derek G. Corneil and Juraj Stacho. “Vertex Ordering Characterizations of Graphs of Bounded Asteroidal Number”. In: *Journal of Graph Theory* 78.1 (2015), pp. 61–79.

[CKL10] Derek G Corneil, Ekkehard Köhler, and Jean-Marc Lanlignel. “On end-vertices of lexicographic breadth first searches”. In: *Discrete Applied Mathematics* 158.5 (2010), pp. 434–443.

[CK08] Derek G Corneil and Richard M Krueger. “A unified view of graph searching”. In: *SIAM Journal on Discrete Mathematics* 22.4 (2008), pp. 1259–1276.

[CC85] G. Cornuéjols and W.H. Cunningham. “Compositions for perfect graphs”. In: *Discrete Mathematics* 55.3 (1985), pp. 245–254.

[CH94] Alain Cournier and Michel Habib. “A new linear algorithm for modular decomposition”. In: *Trees in algebra and programming (CAAP 94)*. Vol. 787. Lecture Notes in Computer Science. 1994, pp. 68–84.

[CS18] H. Perret du Cray and I. Sau. “Improved FPT algorithms for weighted independent set in bull-free graphs”. In: *Discrete Mathematics* 341.2 (2018), pp. 451–462.

[Cun82] William H. Cunningham. “Decomposition of directed graphs”. In: *SIAM Journal on Algebraic and Discrete Methods* 3.2 (1982), pp. 214–228.

[DV13] Murilo VG Da Silva and Kristina Vušković. “Decomposition of even-hole-free graphs with star cutsets and 2-joins”. In: *Journal of Combinatorial Theory, Series B* 103.1 (2013), pp. 144–183.

[DV07] Murilo VG Da Silva and Kristina Vušković. “Triangulated neighborhoods in even-hole-free graphs”. In: *Discrete Mathematics* 307.9-10 (2007), pp. 1065–1073.

[Dab12] K. Dabrowski. “Structural Solutions to Maximum Independent Set and Related Problems”. PhD thesis. University of Warwick, 2012.

[Dab+12] K. Dabrowski, V. V. Lozin, H. Müller, and D. Rautenbach. “Parameterized complexity of the weighted independent set problem beyond graphs of bounded clique number”. In: *J. Discrete Algorithms* 14 (2012), pp. 207–213.

[Dah94] E. Dahlhaus. "Efficient Parallel and Linear Time Sequential Split Decomposition (Extended Abstract)". In: *Proceedings of the 14th Conference on Foundations of Software Technology and Theoretical Computer Science*. London, UK: Springer-Verlag, 1994, pp. 171–180.

[Dah00a] E. Dahlhaus. "Parallel Algorithms for Hierarchical Clustering and Applications to Split Decomposition and Parity Graph Recognition". In: *J. Algorithms* 36.2 (2000), pp. 205–240.

[Dah00b] Elias Dahlhaus. "Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition". In: *Journal of Algorithms* 36.2 (2000), pp. 205–240.

[Dan86] L Danzer. "Zur Lösung des Gallaischen Problems über Kreisscheiben in der Euklidischen Ebene". In: *Studia Sci. Math. Hungar* 21.1-2 (1986), pp. 111–134.

[Déf06] David Défossez. "Clique-coloring some classes of odd-hole-free graphs". In: *Journal of Graph Theory* 53.3 (2006), pp. 233–249.

[Des54] Blanche Descartes. "Solution to Advanced Problem No. 4526". In: *Amer. Math. Monthly* 61 (1954), p. 352.

[DF13] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[DH18] François Dross and Frédéric Havet. "On the unavoidability of oriented trees". In: *arXiv preprint arXiv:1812.05167* (2018).

[DGR95] Dwight Duffus, Mark Ginn, and Vojtech Rödl. "On the Computational Complexity of Ordered Subgraph Recognition". In: *Random Struct. Algorithms* 7.3 (1995), pp. 223–268.

[Duf+91] Dwight Duffus, Bill Sands, Norbert Sauer, and Robert E Woodrow. "Two-colouring all two-element maximal antichains". In: *Journal of Combinatorial Theory, Series A* 57.1 (1991), pp. 109–116.

[DH17] Jérémie Dusart and Michel Habib. "A new LBFS-based algorithm for cocomparability graph recognition". In: *Discrete Applied Mathematics* 216 (2017), pp. 149–161.

[Erd59] Paul Erdos. "Graph theory and probability". In: *canad. J. Math* 11.11 (1959), pp. 34–38.

[Erd79] Paul Erdös. "Problems and results in number theory". In: *Proc. Ninth Manitoba Conference on Numerical Math. and Computing*. 1979, pp. 3–21.

[EGP66] Paul Erdos, Adolph W Goodman, and Lajos Pósa. "The representation of a graph by set intersections". In: *Canad. J. Math* 18.106-112 (1966), p. 86.

[EJS05] Thomas Erlebach, Klaus Jansen, and Eike Seidel. "Polynomial-Time Approximation Schemes for Geometric Intersection Graphs". In: *SIAM J. Comput.* 34.6 (2005), pp. 1302–1323.

[Eve+03] Guy Even, Zvi Lotker, Dana Ron, and Shakhar Smorodinsky. "Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks". In: *SIAM Journal on Computing* 33.1 (2003), pp. 94–136.

[EI71] Shimon Even and Alon Itai. "Queues, stacks, and graphs". In: *Theory of Machines and Computations*. Ed. by Zvi Kohavi and Azaria Paz. Academic Press, 1971, pp. 71–86.

[FOS11] Yuri Faenza, Gianpaolo Oriolo, and Gautier Stauffer. "An algorithmic decomposition of claw-free graphs leading to an O (n 3)-algorithm for the weighted stable set problem". In: *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2011, pp. 630–646.

[Far89] Martin Farber. "On diameters and radii of bridged graphs". In: *Discrete Mathematics* 73.3 (1989), pp. 249–260.

[FH18] Laurent Feuilloley and Michel Habib. "Graph classes and forbidden patterns on three vertices". In: *arXiv preprint arXiv:1812.05913* (2018).

[Fia+12] Jiří Fiala, Marcin Kamiński, Bernard Lidickỳ, and Daniël Paulusma. "The k-in-a-path problem for claw-free graphs". In: *Algorithmica* 62.1-2 (2012), pp. 499–519.

[Fis03] Aleksei V. Fishkin. "Disk Graphs: A Short Survey". In: *Approximation and Online Algorithms, First International Workshop, WAOA 2003, Budapest, Hungary, September 16-18, 2003, Revised Papers*. Ed. by Klaus Jansen and Roberto Solis-Oba. Vol. 2909. Lecture Notes in Computer Science. Springer, 2003, pp. 260–264.

[GHS89] Csaba P. Gabor, Wen-Lian Hsu, and Kenneth J. Supowit. "Recognizing circle graphs in polynomial time". In: *Journal of ACM* 36.3 (1989), pp. 435–473.

[Gal68] Tibor Gallai. "On directed paths and circuits". In: *Theory of graphs* (1968), pp. 115–118.

[GP03] C. Gavoille and C. Paul. "Distance labeling scheme and split decomposition". In: *Discrete Mathematics* 273 (2003), pp. 115–130.

[Gho62] A Ghouilà-Houri. "Characterization of nonoriented graphs whose edges can be oriented in such a way as to obtain the graph of an order relation". In: *CR Acad. Sci. Paris* 254 (1962), pp. 1370–1371.

[GP10] Matt Gibson and Imran A. Pirwani. "Algorithms for Dominating Set in Disk Graphs: Breaking the log$n$ Barrier - (Extended Abstract)". In: *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I.* 2010, pp. 243–254.

[GP07] E. Gioan and C. Paul. "Dynamic Distance Hereditary Graphs Using Split Decomposition". In: *Algorithms and Computation, 18th International Symposium (ISAAC)*. 2007, pp. 41–51.

[Gio+14a] Emeric Gioan, Christophe Paul, Marc Tedder, and Derek Corneil. "Practical and efficient circle graph recognition". In: *Algorithmica* 69.4 (2014), pp. 759–788.

[Gio+14b] Emeric Gioan, Christophe Paul, Marc Tedder, and Derek Corneil. "Practical and Efficient Split Decomposition via Graph-Labelled Trees". In: *Algorithmica* 69.4 (Aug. 2014), pp. 789–843.

[Gol14] Noah Golowich. "Acyclic Colorings and Subgraphs of Directed Graphs". PhD thesis. Massachusetts Institute of Technology, 2014.

[Gol04] Martin Charles Golumbic. “Algorithmic graph theory and perfect graphs”. In: *Annals of Discrete Mathematics*. Second. Vol. 57. Elsevier, 2004, p. 314.

[GSW98] Albert Gräf, Martin Stumpf, and Gerhard Weißenfels. “On coloring unit disk graphs”. In: *Algorithmica* 20.3 (1998), pp. 277–293.

[GHM03] Sylvain Gravier, Chı nh T Hoàng, and Frédéric Maffray. “Coloring the hypergraph of maximal cliques of a graph with no long path”. In: *Discrete mathematics* 272.2-3 (2003), pp. 285–290.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[GLS81] Martin Grötschel, László Lovász, and Alexander Schrijver. “The ellipsoid method and its consequences in combinatorial optimization”. In: *Combinatorica* 1.2 (1981), pp. 169–197.

[Grz+17] A. Grzesik, T. Klimosova, M. Pilipczuk, and M. Pilipczuk. “Polynomial-time algorithm for Maximum Weight Independent Set on P6-free graphs”. In: *CoRR* abs/1707.05491 (2017).

[Gyá75] A Gyárfás. “On Ramsey covering-numbers”. In: *Infinite and Finite Sets* 2 (1975), pp. 801–816.

[Gyá90] A Gyárfás. “Problem 115”. In: *Discrete Mathematics* 79 (1990), pp. 109–110.

[Gyá87] András Gyárfás. “Problems from the world surrounding perfect graphs”. In: *Applicationes Mathematicae* 19.3-4 (1987), pp. 413–441.

[GST80] András Gyárfás, Endre Szemerédi, and Zsolt Tuza. “Induced subtrees in graphs of large chromatic number”. In: *Discrete Mathematics* 30.3 (1980), pp. 235–244.

[HMD14] M. Habib, A. Mamcarz, and F. De Montgolfier. “Computing H-joins with application to 2-modular decomposition”. English. In: *Algorithmica* 70.2 (2014). Cited By :1, pp. 245–266.

[Hab+00] Michel Habib, Ross McConnell, Christophe Paul, and Laurent Viennot. “Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing”. In: *Theoretical Computer Science* 234.1-2 (2000), pp. 59–84.

[HM17a] Michel Habib and Lalla Mouatadid. “Maximum induced matching algorithms via vertex ordering characterizations”. In: *arXiv preprint arXiv:1707.01245* (2017).

[HP10] Michel Habib and Christophe Paul. “A survey of the algorithmic aspects of modular decomposition”. In: *Computer Science Review* 4.1 (2010), pp. 41–59.

[Har+18] Sariel Har-Peled, Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, Micha Sharir, and Max Willert. “Stabbing pairwise intersecting disks by five points”. In: *CoRR* abs/1801.03158 (2018).

[Har+17] Ararat Harutyunyan, Tien-Nam Le, Alantha Newman, and Stéphan Thomassé. “Coloring dense digraphs”. In: *Electronic Notes in Discrete Mathematics* 61 (2017), pp. 577–583.

[HM11a] Ararat Harutyunyan and Bojan Mohar. “Gallai’s theorem for list coloring of digraphs”. In: *SIAM Journal on Discrete Mathematics* 25.1 (2011), pp. 170–180.

[HM17b] Ararat Harutyunyan and Bojan Mohar. “Planar digraphs of digirth five are 2-colorable”. In: *Journal of Graph Theory* 84.4 (2017), pp. 408–427.

[HM11b] Ararat Harutyunyan and Bojan Mohar. “Strengthened Brooks’ theorem for digraphs of girth at least three”. In: *the electronic journal of combinatorics* 18.1 (2011), p. 195.

[HM12] Ararat Harutyunyan and Bojan Mohar. “Two results on the digraph chromatic number”. In: *Discrete Mathematics* 312.10 (2012), pp. 1823–1826.

[Has65] Maria Hasse. “Zur algebraischen begründung der graphentheorie. i”. In: *Mathematische Nachrichten* 28.5-6 (1965), pp. 275–290.

[HW86] D. Haussler and E. Welzl. “Epsilon-Nets and Simplex Range Queries”. In: (1986), pp. 61–71.

[HMR14] Pavol Hell, Bojan Mohar, and Arash Rafiey. “Ordering without Forbidden Patterns”. In: *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*. 2014, pp. 554–565.

[HK01] Petr Hliněný and Jan Kratochvíl. “Representing graphs by disks and balls (a survey of recognition-complexity results)”. In: *Discrete Mathematics* 229.1 (2001), pp. 101–124.

[HM85] Dorit S. Hochbaum and Wolfgang Maass. “Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI”. In: *J. ACM* 32.1 (Jan. 1985), pp. 130–136.

[Hol81] Ian Holyer. *The NP-Completeness of Edge-Colouring*. 1981.

[HM03] Wen-Lian Hsu and Ross M. McConnell. “PC-trees and circular-ones arrangements”. In: *Theoretical Computer Science* 296 (2003), pp. 99–116.

[JH] Ramin Javadi and Sepehr Hajebi. “Edge clique cover of claw-free graphs”. In: *Journal of Graph Theory* 0.0 ().

[KM] Gil Kalai and Roy Meshulam. *https://gilkalai.files.wordpress.com/2010/10/es.pdf*.

[KM12] Ross J. Kang and Tobias Müller. “Sphere and Dot Product Representations of Graphs”. In: *Discrete & Computational Geometry* 47.3 (2012), pp. 548–568.

[Kar17] T. Karthick. “Independent Sets in Some Classes of $S_{i,j,k}$-free Graphs”. In: *J. Comb. Optim.* 34.2 (Aug. 2017), pp. 612–630.

[KM17] T. Karthick and F. Maffray. “Maximum weight independent sets in classes related to claw-free graphs”. In: *Discrete Applied Mathematics* 216 (2017), pp. 233–239.

[KR96] Hal A Kierstead and V Rodl. “Applications of hypergraph coloring to coloring graphs not inducing certain trees”. In: *Discrete Mathematics* 150.1-3 (1996), pp. 187–193.

[KT92] Hal A Kierstead and William T Trotter. “Colorful induced subgraphs”. In: *Discrete mathematics* 101.1-3 (1992), pp. 165–169.

[KZ04] Hal A Kierstead and Yingxian Zhu. “Radius three trees in graphs with large chromatic number”. In: *SIAM Journal on Discrete Mathematics* 17.4 (2004), pp. 571–581.

[KP94] Henry A Kierstead and Stephen G Penrice. "Radius two trees specify χ-bounded classes". In: *Journal of Graph Theory* 18.2 (1994), pp. 119–129.

[KG09] A King and Claw-Free Graphs. "Two Conjectures on ω, Δ, and χ". PhD thesis. Ph. D. thesis, McGill University, Montreal, Canada, 2009.

[Koe36] Paul Koebe. "Kontaktprobleme der konformen Abbildung". In: *Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physikalische Klasse* 88 (1936), pp. 141–164.

[KM14] Ekkehard Köhler and Lalla Mouatadid. "Linear Time LexDFS on Cocomparability Graphs." In: *Algorithm Theory – SWAT 2014*. Ed. by R. Ravi and Inge Li Gørtz. Cham: Springer International Publishing, 2014, pp. 319–330.

[KO04] Daniela Kühn and Deryk Osthus. "Induced subdivisions in K s, s-free graphs of large average degree". In: *Combinatorica* 24.2 (2004), pp. 287–304.

[Lee06] Erik Jan van Leeuwen. "Better Approximation Schemes for Disk Graphs". In: *Algorithm Theory - SWAT 2006, 10th ScandinavianWorkshop on Algorithm Theory, Riga, Latvia, July 6-8, 2006, Proceedings*. 2006, pp. 316–327.

[LB62] C Lekkeikerker and J Boland. "Representation of a finite graph by a set of intervals on the real line". In: *Fundamenta Mathematicae* 51.1 (1962), pp. 45–64.

[LMP09] Benjamin Lévêque, Frédéric Maffray, and Myriam Preissmann. "Characterizing path graphs by forbidden induced subgraphs". In: *Journal of Graph Theory* 62.4 (2009), pp. 369–384.

[LVV14] D. Lokshtanov, M. Vatshelle, and Y. Villanger. "Independent Set in P_5-Free Graphs in Polynomial Time". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*. 2014, pp. 570–581.

[Lov72] László Lovász. "Normal hypergraphs and the perfect graph conjecture". In: *Discrete Mathematics* 2.3 (1972), pp. 253–267.

[MS94a] Tze-Heng Ma and Jeremy Spinrad. "An O (n2) algorithm for undirected split decomposition". In: *Journal of Algorithms* 16.1 (1994), pp. 145–160.

[MP96] Frédéric Maffray and Myriam Preissmann. "On the NP-completeness of the k-colorability problem for triangle-free graphs". In: *Discrete Mathematics* 162.1-3 (1996), pp. 313–317.

[MTV08] Frédéric Maffray, Nicolas Trotignon, and Kristina Vušković. "Algorithms for square-3PC (,)-free Berge graphs". In: *SIAM Journal on Discrete Mathematics* 22.1 (2008), pp. 51–71.

[MP15] Dániel Marx and Michal Pilipczuk. "Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams". In: *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*. 2015, pp. 865–877.

[McC04] R.M. McConnell. "A certifying algorithm for the consecutive-ones property". In: *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2004, p. 777.

[MS94b] Ross M. McConnell and Jeremy P. Spinrad. "Linear-time modular decomposition and efficient transitive orientation of comparability graphs". In: *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Arlington, VA: ACM, 1994, pp. 536–545.

[MM99] Terry A McKee and Fred R McMorris. *Topics in intersection graph theory*. SIAM, 1999.

[MC12] G. Mertzios and D. Corneil. "A Simple Polynomial Algorithm for the Longest Path Problem on Cocomparability Graphs". In: *SIAM Journal on Discrete Mathematics* 26.3 (2012), pp. 940–963.

[Moh03] B Mohar. *Choosability for the circular chromatic number*. 2003.

[Moh10] Bojan Mohar. "Eigenvalues and colorings of digraphs". In: *Linear Algebra and its Applications* 432.9 (2010), pp. 2273–2277.

[MS99] Bojan Mohar and Riste Skrekovski. "The Grötzsch theorem for the hypergraph of maximal cliques". In: *Electron. J. Combin* 6.1 (1999), p. 128.

[MW16] Bojan Mohar and Hehui Wu. "Dichromatic Number And Fractional Chromatic Number". In: *Forum of Mathematics, Sigma* 4 (2016), e32.

[Möh85] Rolf H Möhring. "Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions". In: *Annals of Operations Research* 6 (1985), pp. 195–225.

[MR84] Rolf H Möhring and Franz Josef Radermacher. "Substitution decomposition for discrete structures and connections with combinatorial optimization". In: *Annals of Discrete Mathematics* 19 (1984), pp. 257–356.

[Myc55] Jan Mycielski. "Sur le coloriage des graphes". In: *Colloq. Math*. Vol. 3. 161-162. 1955, p. 9.

[Naj85] W. Naji. "Reconnaissance des graphes de cordes". In: *Discrete mathematics* 54.3 (1985), pp. 329–337.

[Neu82] Victor Neumann-Lara. "The dichromatic number of a digraph". In: *Journal of Combinatorial Theory, Series B* 33.3 (1982), pp. 265–270.

[Neu85] Victor Neumann-Lara. "Vertex colourings in digraphs". In: *Some Problems. Seminar notes, University of Waterloo*. 1985.

[NH05] Tim Nieberg and Johann Hurink. "A PTAS for the Minimum Dominating Set Problem in Unit Disk Graphs". In: *Approximation and Online Algorithms, Third International Workshop, WAOA 2005, Palma de Mallorca, Spain, October 6-7, 2005, Revised Papers*. 2005, pp. 296–306.

[NHK04] Tim Nieberg, Johann Hurink, and Walter Kern. "A Robust PTAS for Maximum Weight Independent Sets in Unit Disk Graphs". In: *Graph-Theoretic Concepts in Computer Science, 30th International Workshop,WG 2004, Bad Honnef, Germany, June 21-23, 2004, Revised Papers*. 2004, pp. 214–221.

[Ola91] Stephan Olariu. "An Optimal Greedy Heuristic to Color Interval Graphs". In: *Inf. Process. Lett.* 37.1 (1991), pp. 21–25.

[Oum17] Sang-il Oum. “Rank-width: Algorithmic and structural results”. In: *Discrete Applied Mathematics* 231 (2017). Algorithmic Graph Theory on the Adriatic Coast, pp. 15–24.

[Pau06] Christophe Paul. “Algorithmic aspects of modular decomposition”. Habilitation à diriger des recherches. Université Montpellier II - Sciences et Techniques du Languedoc, July 2006.

[Pen16] Irena Penev. “Perfect Graphs with No Balanced Skew-Partition Are 2-Clique-Colorable”. In: *J. Graph Theory* 81.3 (2016), pp. 213–235.

[Pol74] Svatopluk Poljak. “A note on stable sets and colorings of graphs”. In: *Commentationes Mathematicae Universitatis Carolinae* 15.2 (1974), pp. 307–309.

[RS03] Vijay Raghavan and Jeremy P. Spinrad. “Robust algorithms for restricted domains”. In: *J. Algorithms* 48.1 (2003), pp. 160–172.

[RR88] G. Ramalingam and C.P. Rangan. “A unified approach to domination problems on interval graphs”. In: *Information Processing Letters* 27.5 (1988), pp. 271–274.

[Rao04] M. Rao. “Coloring a Graph Using Split Decomposition”. In: *Graph-Theoretic Concepts in Computer Science (WG)*. Vol. 3353. Lecture Notes in Computer Science. 2004, pp. 129–141.

[Rao08] M. Rao. “Solving some NP-complete problems using split decomposition”. In: *Discrete Applied Mathematics* 156.14 (2008), pp. 2768–2780.

[Röd77] V Rödl. “On the chromatic number of subgraphs of a given graph”. In: *Proceedings of the American Mathematical Society* 64.2 (1977), pp. 370–371.

[RTL76] Donald J. Rose, Robert Endre Tarjan, and George S. Lueker. “Algorithmic Aspects of Vertex Elimination on Graphs”. In: *SIAM Journal on Computing* 5.2 (1976), pp. 266–283.

[Roy67] Bernard Roy. “Nombre chromatique et plus longs chemins d’un graphe”. In: *Revue française d’informatique et de recherche opérationnelle* 1.5 (1967), pp. 129–132.

[Sco97] Alex D Scott. “Induced trees in graphs of large chromatic number”. In: *Journal of Graph Theory* 24.4 (1997), pp. 297–311.

[SS18a] Alex Scott and Paul Seymour. “A survey of χ-boundedness”. In: *arXiv preprint arXiv:1812.07500* (2018).

[SS16] Alex Scott and Paul Seymour. “Induced subgraphs of graphs with large chromatic number. I. Odd holes”. In: *Journal of Combinatorial Theory, Series B* 121 (2016), pp. 68–84.

[SS15] Alex Scott and Paul Seymour. “Induced subgraphs of graphs with large chromatic number. IV. Consecutive holes”. In: *arXiv preprint arXiv:1509.06563* (2015).

[SS17a] Alex Scott and Paul Seymour. “Induced subgraphs of graphs with large chromatic number. IX. Rainbow paths”. In: *arXiv preprint arXiv:1702.01094* (2017).

[SS17b] Alex Scott and Paul Seymour. “Induced subgraphs of graphs with large chromatic number. IX. Rainbow paths”. In: *arXiv preprint arXiv:1702.01094* (2017).

[SS17c] Alex Scott and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. VI. Banana trees". In: *arXiv preprint arXiv:1701.05597* (2017).

[SS17d] Alex Scott and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. VII. Gyárfás' complementation conjecture". In: *arXiv preprint arXiv:1701.06301* (2017).

[SS17e] Alex Scott and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. X. Holes of specific residue". In: (2017).

[SS18b] Alex Scott and Paul Seymour. "Induced subgraphs of graphs with large chromatic number. XIII. New brooms". In: *arXiv preprint arXiv:1807.03768* (2018).

[Sim91] Klaus Simon. "A new simple linear algorithm to recognize interval graphs". In: *Workshop on Computational Geometry*. Springer. 1991, pp. 289–308.

[SW98] Warren D. Smith and Nicholas C. Wormald. "Geometric Separator Theorems & Applications". In: *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*. 1998, pp. 232–243.

[Spi98] Frits C. R. Spieksma. "Approximating an interval scheduling problem". In: *Approximation Algorithms for Combinatiorial Optimization*. Ed. by Klaus Jansen and José Rolim. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 169–180.

[Spi94] Jeremy P. Spinrad. "Recognition of Circle Graphs". In: *Journal of Algorithms* 16.2 (1994), pp. 264–282.

[Sta] Juraj Stacho. *Private communication*.

[Sta81] Lajos Stachó. "A solution of Gallai's problem on pinning down circles". In: *Mat. Lapok* 32.1-3 (1981), pp. 19–47.

[Sum81] David P Sumner. "Subtrees of a graph and chromatic number". In: *The theory and applications of graphs* (1981), pp. 557–576.

[TTV17] Stéphan Thomassé, N. Trotignon, and Kristina Vušković. "A Polynomial Turing-Kernel for Weighted Independent Set in Bull-Free Graphs". In: *Algorithmica* 77.3 (2017), pp. 619–641.

[Tro08] Nicolas Trotignon. "Decomposing Berge graphs and detecting balanced skew partitions". In: *Journal of Combinatorial Theory, Series B* 98.1 (Jan. 2008), pp. 173–225.

[Tro13] Nicolas Trotignon. "Perfect graphs: a survey". In: *ArXiv e-prints* (2013).

[Tro18] Nicolas Trotignon. "Personal Communication". 2018.

[TV12] Nicolas Trotignon and Kristina Vušković. "Combinatorial optimization with 2-joins". In: *Journal of Combinatorial Theory, Series B* 102.1 (2012), pp. 153–185.

[Tru82] Klaus Truemper. "Alpha-balanced graphs and matrices and GF (3)-representability of matroids". In: *Journal of Combinatorial Theory, Series B* 32.2 (1982), pp. 112–139.

[VC15] Vladimir N Vapnik and A Ya Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities". In: *Measures of complexity*. Springer, 2015, pp. 11–30.

[VA99] Bram Verweij and Karen Aardal. "An Optimisation Algorithm for Maximum Independent Set with Applications in Map Labelling". In: *Algorithms - ESA' 99*. Ed. by Jaroslav Nešetřil. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 426–437.

[Vit62] LM Vitaver. "Determination of minimal coloring of vertices of a graph by means of Boolean powers of the incidence matrix". In: *Dokl. Akad. Nauk SSSR*. Vol. 147. 758-759. 1962, p. 728.

[Vuš10] Kristina Vušković. "Even-hole-free graphs: a survey". In: *Applicable Analysis and Discrete Mathematics* (2010), pp. 219–240.

[Vuš13] Kristina Vušković. "The world of hereditary graph classes viewed through Truemper configurations". In: *Surveys in Combinatorics 2013* 409 (2013), p. 265.

[WJ+07] Kolen Antoon W.J., Lenstra Jan Karel, Papadimitriou Christos H., and Spieksma Frits C.R. "Interval scheduling: A survey". In: *Naval Research Logistics (NRL)* 54.5 (2007), pp. 530–543.

[Woo04] David R. Wood. "Characterisations of Intersection Graphs by Vertex Orderings". In: *CoRR* cs.DM/0404031 (2004).

[Wro18] Martin Wrochna. *Personal Communication*. 2018.

[Zuc07] D. Zuckerman. "Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number". In: *Theory of Computing* 3.1 (2007), pp. 103–128.

[Zyk49] Alexander Aleksandrovich Zykov. "On some properties of linear complexes". In: *Matematicheskii sbornik* 66.2 (1949), pp. 163–188.

Index