

PH.D THESIS

présentée au Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier pour
obtenir le diplôme de doctorat

Spécialité : **Informatique**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

Hitting sets: VC-dimension and Multicut

par

Nicolas BOUSQUET

December, 09, 2013

Supervisor

M. Stéphane BESSY Maître de Conférences, LIRMM, Université Montpellier 2

Joint supervisor

M. Stéphan THOMASSÉ Professeur, LIP, ENS Lyon

Reviewers

M. Bruce REED Professeur, McGill University, Montréal

M. Yann VAXÈS Professeur, LIF, Université d'Aix-Marseille

Examinators

M. Pierre FRAIGNIAUD Directeur de recherche CNRS, LIAFA, Université Paris-Diderot

M. Gwenaël RICHOMME Professeur, LIRMM, Université Montpellier 3

M. Andrés SEBŐ Directeur de recherche CNRS, Laboratoire GSCOP



Contents

Contents	i
Remerciements	1
Introduction	3
Outline of the thesis	6
Publications	9
1 Preliminaries	13
1.1 Hypergraphs	13
1.2 Graphs	16
1.3 Parameterized complexity	22
2 Hitting sets	29
2.1 Hitting sets and packings	29
2.1.1 Definitions and first properties	29
2.1.2 Helly property	33
2.1.3 Duality of hypergraphs	36
2.2 Linear programming and integrality gap	37
2.2.1 Generalities	37
2.2.2 Transversal and Packing Linear Programs	39
2.2.3 Farkas' Lemma	43
2.3 Algorithmic aspects of hitting sets. Application to MULTICUT	44
2.3.1 Multicut problems	45
2.3.2 Important separators	47
2.3.3 Application to MULTIWAY CUT	54
2.3.4 Shadow removal technique	55
2.3.5 Application to DIRECTED MULTIWAY CUT	60

3	VC-dimension	63
3.1	Shattered sets	64
3.1.1	Opening properties	64
3.1.2	First examples	69
3.2	VC-dimension and number of hyperedges	70
3.2.1	The Sandwich Theorem and Sauer's Lemma	71
3.2.2	Ordered shattered sets	75
3.2.3	Application to identifying codes	78
3.3	VC-dimension and integrality gap	79
3.3.1	Upper bound on the size of epsilon-nets	80
3.3.2	Applications	82
3.4	Erdős-Pósa property	86
3.4.1	2VC-dimension	86
3.4.2	Application: Scott's Conjecture for maximal triangle-free graphs	88
3.4.3	VC-dimension and (p,q)-property	91
3.5	Algorithmic aspects of VC-dimension	92
4	Domination at large distance	95
4.1	Introduction	95
4.2	Graphs of bounded VC-dimension	97
4.2.1	K_d -minor-free graphs have bounded VC-dimension	97
4.2.2	Bounded rankwidth graphs have bounded VC-dimension	98
4.2.3	Unbounded 2VC-dimension but bounded VC-dimension	100
4.3	Erdős-Pósa property	102
4.3.1	A lower bound for the VC-dimension of a graph	103
4.3.2	Sparse sets	105
4.3.3	Localized and independent pairs	107
4.3.4	Escape property	109
4.3.5	Escape property implies large VC-dimension	112
4.4	Conclusion	114
5	Separate cliques and stable sets	117
5.1	Introduction	117
5.2	Clique-Stable Set separation conjecture	120
5.2.1	Random graphs	122
5.2.2	The case of split-free graphs	123
5.2.3	The case of P_k, \overline{P}_k -free graphs	125
5.2.4	Other classes of graphs	128
5.3	Bipartite packing and graph coloring	128
5.3.1	Oriented Alon-Saks-Seymour conjecture	128
5.3.2	Generalization: t -biclique covering numbers	132
5.4	3-CCP and the stubborn problem	133
6	Multicut is FPT	141

6.1	Detailed outline of the proof	141
6.1.1	Reductions, branchings and invariants	143
6.1.2	Irrelevant requests	144
6.1.3	Cherry reduction	146
6.2	Reducing MULTICUT to COMPONENT MULTICUT	148
6.2.1	COMPONENT MULTICUT	148
6.2.2	The Vertex-Multicut Y	149
6.2.3	Reducing attachment vertices	150
6.2.4	Backbones	151
6.3	BACKBONE MULTICUT is FPT	152
6.3.1	BACKBONE MULTICUT	152
6.3.2	Invariants	153
6.3.3	Contracting edges	154
6.3.4	Choosing a stable edge	155
6.3.5	Contracting slices	156
6.3.6	Reducing the lemons	159
6.3.7	Reducing to 2-SAT	163
6.4	Hints for VERTEX MULTICUT	165
6.5	Conclusion	166
6.5.1	A single-exponential algorithm	166
6.5.2	Comparison with Marx and Razgon's Proof in [147]	167
6.5.3	Other leads	167
7	Conclusion	169
	Bibliography	173
	Bibliography	173
	Index	185
	List of Figures	188



Remerciements

Je voudrais tout d'abord remercier Bruce Reed et Yann Vaxès pour la relecture attentive de mon manuscrit. Je voudrais aussi remercier les autres membres du jury, Pierre Fraigniaud, Gwenaël Richomme et András Sebő, de s'être déplacés pour ma soutenance.

Je voudrais ensuite remercier les Stéphan(e)s sans qui cette thèse n'aurait pas pu voir le jour. Lors de mon premier stage, je suis venu au LIRMM pour retrouver le soleil montpelliérain, la seconde fois je suis venu à cause de vous. Stéphan, merci de m'avoir proposé un sujet de stage puis de thèse et d'avoir partagé avec moi ta culture combinatoire encyclopédique (et ton amour inconditionnel de l'administratif). Stéphane, merci d'avoir accepté de m'encadrer malgré la "fuite" de mon premier directeur et pour toutes des séances de psy (ou les discussions rugby) alors que tu avais très probablement mieux à faire (même si je n'en suis pas certain dans le second cas !). Vous avez su me donner des conseils avisés, tant sur la forme que sur le fond de mes recherches, en me laissant systématiquement libre de mes choix finaux. Je vous remercie pour tout ce que ces discussions m'ont apporté. Cela m'a permis de travailler sereinement et sur un large éventail de problèmes pour découvrir la richesse de la théorie des graphes.

J'ai eu l'occasion au cours de la thèse de collaborer ou discuter avec de nombreux chercheurs de tous horizons. Je veux les remercier pour les discussions fructueuses que j'ai pu avoir avec eux. Tout d'abord merci aux membres du LIP pour leur accueil à chacune de mes visites, en particulier Nicolas, Sébastien, Théo et bien sûr Aurélie, ma petite soeur de thèse. Mais aussi à ceux d'ailleurs, en particulier Hervé (alias Jean-Bernard), Pierre qui m'a traité de fou (à raison) pour les 3-cordes, Arnaud, avec qui j'ai partagé un certain nombre de bières.

Une thèse ne serait rien sans le cadre de travail, et le LIRMM a été, à tout point de vue, un endroit parfait pour ces trois années. Je voudrais d'abord remercier l'équipe ALGCo de son accueil. Merci d'avoir répondu à toutes mes questions, des plus pertinentes aux plus stupides, de m'avoir motorisé tous les jours pour aller au restaurant du CNRS, d'avoir organisé des repas d'équipe aux sources du Lez, au Crès ou en 3.23 (on a le meilleur repas de Noël du labo, Stéphane tu es un vrai chef !). Une mention toute particulière à Daniel et Alex que j'ai tour à tour "virés" de leur bureau ainsi qu'à Christophe et Ignasi, mes experts ès complexité paramétrée, et à Pascal pour son cours

particulier d'algorithmique des mots.

Comme les chercheurs que nous sommes ne seraient rien si des gens avec les pieds sur terre n'étaient pas là pour les aider, je ne peux que remercier Caroline, Laurie et Nicolas pour leur travail efficace, leur bonne humeur et leur aide précieuse dans des démarches administratives (trop souvent) obscures.

Dans les bons comme les mauvais moments, rien ne serait pareil si d'autres doctorants ne partageaient pas notre quotidien. J'en oublierai sûrement, mais, dans le désordre, Rémi (dont j'ai pas mal squatté le bureau), Guillaume (un jour je comprendrai les cours-métrages dans lesquels je tourne), Mika (qui finit toujours par me suivre), Flavien, Fab' et tous les membres du Conseil des Doctorants. Je veux aussi remercier plus particulièrement les doctorants de l'équipe AIGCo pour avoir supporté mes grognements (il paraît que je tiens ça d'un de mes directeurs). Évidemment j'ai quand même réussi à en faire fuir certains: Antho (le plus montpelliérain des nîmois) et Jean (spécialiste ès jeu), qui m'ont guidé au début de ma thèse. Merci aussi à ceux qui n'ont pas fui: Guillaume, chevalier de l'esperluette; Valentin et son drapeau polonais; Boris, pour m'avoir fait lire le magazine d'Air France comme personne. Et bien sûr Marthe qui m'a nourri (et supporté) presque tous les jours à partir de 16h30 du matin et Kevin, Sarah et Malah qui ont toujours été là même dans les moments les plus difficiles (Kevin, je sais pas si on peut dire que je t'ai fait fuir comme ça s'est joué à une semaine près).

Rien ne serait possible sans des amis pour nous épauler tout au long du parcours. A toute la bande des joyeux matheux cachanais que je préviens (paraît-il) toujours au dernier moment, Pierre, Sarah, Cricri, Flo et Xanou et leur canapé, Julot, Jean (magnifique en princesse) et bien sûr Alice et Hoel (rendez-vous en Amérique). Mais aussi aux infos (A/B0), Luc, Max, Alex, Élie et Cie. A la compagnie "que rien n'arrête" qui m'a valu quelques nuits blanches (et pas mal de fous rires), FLO, Anna, Romain ("tiens, c'est marrant, on est 4 gauchers."), Pascal (un droitier, personne n'est parfait) et tous les autres. A tous les ingénieurs (et architectes) que Sandrine m'a fait côtoyer, surtout les bretons. Avec une mention toute particulière pour PA qui m'a supporté comme coloc' improvisé dans une ville néerlandaise que je suis incapable d'épeler correctement.

Merci enfin à ma famille et à ma belle-famille de m'épauler tous les jours. A Jean-Guy qui a (presque) fait de moi un bricoleur, Marie-Christine pour de multiples discussions de chercheurs, sans oublier Jérôme. A Arnault qui a perdu de son temps précieux de taupin pour m'aider en maths (ça n'aura pas été vain). A Romain et Caro qui m'hébergent toujours avec gentillesse à Lyon (avec des super tartes). Et bien sûr à mes parents qui nous accueillent toujours à bras ouverts même quand nous les prévenons (eux-aussi) au dernier moment. Merci à tous pour votre aide et votre soutien en toutes circonstances. Un grand merci aussi à tous ceux qui ont fait le déplacement à ma soutenance en sachant très bien qu'ils n'allaient (presque) rien comprendre.

Enfin, cette thèse n'aurait bien sûr pas pu voir le jour sans Sandrine. Merci de me supporter tous les jours avec toutes les contraintes de la recherche (il faut pas mal de courage). Merci de partager avec moi les petits tracas et les grandes joies du quotidien. Merci pour ce jour fabuleux qu'a été le 22 Juin. Une page se tourne, une nouvelle s'ouvre, allons l'écrire ensemble aux quatre coins du monde!



Introduction

Context. In the last decades, the importance of computer science grew in the everyday life. In the same time, theoretical aspects of computer science have been developed, and in particular algorithmics. The aim of algorithmics is to find the best algorithm for solving a problem, where best means that it both provides the best solution and finds it faster than other algorithms. Algorithmic theory provides some tools for designing algorithms or for proving that some problems cannot be efficiently solved. At the beginning of computer science, algorithms were often very simple. Today they are more and more involved as the sizes of the problems increase (more tasks are given to computers and the amount of data to treat is ever increasing). Mathematics, and particularly combinatorics, provide useful tools for designing algorithms and finding (positive or negative) bounds on their complexity. Indeed, mathematical structures can lead to algorithms that make use of deeper concepts to be much more efficient.

Since computers only deal with finite structures, the notions of graphs and hypergraphs and more generally all the combinatorial notions, arise naturally in various fields of computer science. A graph is a set of points called vertices where some pairs of vertices are connected by links called edges. For instance, a road network can be represented by a graph where the vertices of the graph are crossings of the network and edges of the graphs are the existing roads between crossings. Hypergraphs generalize the notion of graphs since, instead of connecting pairs of vertices with links, we can create relations between arbitrarily many vertices, with so-called hyperedges. From an applicative point of view, the concept of graphs is sometimes not adequate. For instance hypergraphs can represent social networks better than graphs. In this case, the vertices of the hypergraph represent people and hyperedges represent a point of mutual interest shared by the set of persons (for instance they practice the same activity). Keep in mind that edges represent binary relations (the vertex a is linked with the vertex b) while hyperedges represent relations of an arbitrarily large size (the whole set of vertices A has the same property).

During my PhD, I was interested in a particular problem on hypergraphs: the HITTING SET problem. I studied it from both a combinatorial and an algorithmic point of view using VC-dimension and important separators. A hitting set is a set of vertices which intersects every hyperedge. In other

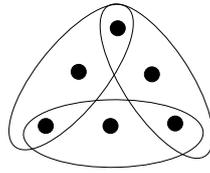


Figure 0.1: Non equality between minimum hitting set and maximum packing.

words, it is a subset of vertices such that every hyperedge contains at least one vertex of the hitting set. A packing is a set of vertex-disjoint hyperedges. In other words, no vertex appears in two hyperedges of a packing. The size of a hitting set denotes its number of vertices while the size of a packing denotes its number of hyperedges. The HITTING SET problem consists in, given a hypergraph H and an integer k , determining if H has a hitting set of size at most k . The PACKING problem consists in, given a hypergraph H and an integer k , determining if H has a packing of size at least k . PACKING and HITTING SET problems naturally generalize many problems on graphs and hypergraphs such as VERTEX COVER, MAXIMUM MATCHING, 3D-MATCHING, CHROMATIC NUMBER, DOMINATING SET, MULTICUT, IDENTIFYING CODE, FEEDBACK VERTEX SET.

The size of a hitting set is at least the size of a packing. Indeed, no vertex appears in two hyperedges of a packing. So, if we want a set of vertices intersecting all the hyperedges, we need pairwise distinct vertices for the hyperedges of the packing. Hence the minimum size of a hitting set denoted by τ is at least the maximum size of a packing denoted by ν , *i.e.* we have $\tau \geq \nu$. In the general case this inequality is not an equality. In Figure 0.1, the equality between τ and ν is not achieved. Indeed, every pair of hyperedges intersects; so the maximum size of a packing is one. On the contrary, since no vertex is in the intersection of the three hyperedges, the minimum size of a hitting set is at least two (and actually exactly two).

There exists a very simple way to determine if an instance of HITTING SET is satisfied. Denote by H the hypergraph and by k the integer of the instance. For every subset of vertices of size k , check if this subset is a hitting set or not. If at least one subset of size k is a hitting set, then the answer is positive. Otherwise, it is negative. However, the “problem” of this algorithm is its cost. Indeed, if we are looking for hitting sets of size 6 in hypergraphs of size 1000 (a “small” instance), then the algorithm will check if a billion of billion of sets are hitting sets or not. So it is a main issue to find more efficient algorithms even for subcases of the HITTING SET problem.

This naturally raises two questions on hitting sets:

- (1) We have seen that $\tau \geq \nu$. Can we “reverse” this inequality? More precisely does there exist a function f such that $\tau \leq f(\nu)$? Or if the answer is negative in general hypergraphs, what conditions ensure that there exists such a function?
- (2) We have seen an algorithm for deciding the HITTING SET problem which seems not really efficient. Can we find “efficient algorithms” in order to compute τ and ν ? Or if the answer is negative in general hypergraphs, what conditions on the hypergraph ensure that we can efficiently evaluate these values?

The goal of this manuscript is to provide some (partial) answers to both questions using in particular two general tools: the VC-dimension and the important separators.

Gap between τ and ν and VC-dimension. The answer to the first question has been well known for decades: there is no function linking τ and ν . Nevertheless, for dozens of classes of hypergraphs there exist functions linking τ and ν . Historically, one of the first (and most famous) bound between τ and ν is due to Erdős and Pósa who proved that for any graph the gap between the maximum number of vertex-disjoint cycles and the minimum number of vertices intersecting all the cycles is bounded. In other words they proved that there is a bounded gap between τ and ν in the so-called cycle hypergraphs. When the gap between τ and ν is bounded for a class of hypergraphs, we say that the class satisfies the Erdős-Pósa property.

So, in order to obtain the Erdős-Pósa property, we have to add constraints on the hyperedges. For instance, we can add size constraints (the size of every hyperedge is bounded), or geometrical constraints (hyperedges can represent rectangles, or lines in a geometrical space). Over all the existing invariants on hypergraphs, one is particularly useful for finding (upper) bounds on the size of hitting sets: the dimension of Vapnik-Chervonenkis (VC-dimension for short). The VC-dimension is a complexity measure on hypergraphs. A set of vertices is shattered if the hyperedges intersect it in all possible ways. More formally, a set X is shattered if for every $X' \subseteq X$ there exists a hyperedge e such that $e \cap X = X'$. The VC-dimension is the maximum size of a shattered set. The main statement on VC-dimension ensures that τ can be bounded by a function of the so-called fractional relaxation of τ and the VC-dimension. In other words, when the VC-dimension is bounded, τ is bounded by a function of its fractional relaxation, while this is false on general hypergraphs. In addition, under stronger conditions on the VC-dimension, the gap between τ and ν can also be bounded.

One of the main objectives of my PhD was to study the VC-dimension machinery in order to understand it and apply it on hypergraphs, but also on graphs.

Algorithmic aspects of hitting sets and important separators. Before looking more into to the second question, let us first give some informal definitions on complexity and try to define a little bit more properly what “efficient algorithm” means. When we are given an instance of a problem (understand a “question”), we want to find a solution (understand an “answer”). But we want some conditions on this answer. First we want a correct answer: if somebody asks a question, he is waiting for a good answer. But there is a second crucial point: the time complexity. If somebody asks a question, he does not want to wait the answer for decades (nobody likes his computer to lag).

Historically, the first definition of “efficient” algorithm in theoretical computer science is an algorithm running in polynomial time. The research of polynomial time algorithm for solving problems was and is still one of the main goals of algorithmic graph theory. However, many problems seem to not admit polynomial time algorithms. This intuition was formalized by Cook in the 70’s. He defined a complexity class called NP which is “the class of problems which can be solved in polynomial time with a non-deterministic Turing machine”. We will not properly define the class NP (which needs technical definitions) but, roughly, a problem is in NP if solutions can be verified in polynomial time (you can efficiently check if the solver lied to you). The HITTING SET problem is in NP . Indeed, if you are given a set of vertices, you can check in polynomial time if the set intersects every hyperedge or not. Similarly the PACKING PROBLEM is in NP . Cook proved that a problem of NP is at least as difficult as the other problems of NP : the SAT problem. All the problems of NP which are as difficult as SAT are said to be NP-complete. He finally conjectured the following:

Conjecture 1 ($P \neq NP$). *There is no polynomial time algorithm for deciding the SAT problem.*

Conjecture 1 is still open today and is considered as the main open problem of theoretical computer science. In the remaining, we assume that Conjecture 1 holds. Since Cook's paper, thousands of problems have been proved to be NP-complete, and then (probably) do not admit polynomial time algorithms. Unfortunately, both PACKING and HITTING SET problems are NP-complete. So we cannot (under Conjecture 1) compute in polynomial time a hitting set of minimum size (nor a packing of maximum size) on general hypergraphs.

Since, several methods for designing as efficient as possible algorithms for NP-complete problems have been developed. Approximation algorithms are polynomial time algorithms which are looking for non-necessarily optimal solution but solutions with some guarantees compared to the optimal ones. A second issue consists in finding exponential-time algorithms with exponents which are as small as possible. One can also develop heuristics which are not polynomial in theory but efficient in practice (SAT solvers for instance). In this manuscript we only deal with another field of theoretical algorithmic theory: the parameterized complexity.

The idea of the parameterized complexity is the following: an NP-complete problem cannot be solved in polynomial time, so the goal is to confine the combinatorial explosion of the algorithm to a small parameter instead of the whole size of the instance. So the aim is to develop an algorithm which is polynomial in the size of the input except for this small parameter. More formally, a problem is FPT (Fixed Parameter Tractable) according to a parameter k if there exists an algorithm that decides any instance of size n in time $f(k)n^c$ where c is a constant and f a computable function. If this parameter is small enough, the resulting algorithm can be efficient even if its complexity is not polynomial. More precisely, there exists a polynomial time algorithm for every fixed k whose power does not depend on k .

During my PhD, I studied several hitting set problems from a parameterized point of view. In particular I focused on the MULTICUT problem using a notion defined in the last few years: important separators. Important separators are particular separators which can be favored compared to the others (whenever we want to minimize a connected component). Marx developed this tool and proved that in many graph separation problems an important separator has to be selected. In addition he proved that the number of important separators of size k is bounded. More precisely, the number of important separators of size at most k between x and y can be bounded by a function of k and all of them can be enumerated in time $f(k)$ (which indeed is interesting for designing FPT algorithms).

Outline of the thesis

Chapters 1, 2 and 3 are general chapters which are devoted to introducing and explaining all the tools used along the manuscript. Unless specified otherwise, I did not participate to the proofs of the results mentioned along these three chapters. Chapter 4, 5 and 6 are devoted to a presentation of some of the results obtained during my PhD.

In Chapter 1, we provide classical definitions on graphs and hypergraphs. We also recall well-known theorems used throughout the manuscript. We end this chapter by introducing the parameterized complexity and by looking a little bit further at the notions of FPT algorithms and kernel algorithms.

In Chapter 2, we introduce the two main notions of this manuscript: hitting sets and packings. We see that lots of invariants on graphs can be viewed as particular cases of hitting sets, such as vertex covers or dominating sets. We also define more properly the transversality τ and the packing number ν and we study several classes of hypergraphs satisfying $\tau \leq f(\nu)$, but also classes such that the gap between them is arbitrarily large in the general case. In a second part, we introduce notions of linear programming and prove that both τ and ν can be expressed as objective functions of integer linear programs. We then introduce the fractional relaxation of the transversality, denoted by τ^* , whose value is between τ and ν . The fractional transversality will be a crucial notion for finding upper bounds on the transversality in Chapter 3.

We finally deal with graph separation problems which are particular cases of HITTING SET problems. We introduce the MULTICUT problem; and we study in details two important methods recently developed for designing algorithms on graph separation problems. The first one, due to Marx, is called the important separators technique. It consists in finding separators which can be favored compared to other separators. The second one, due to Marx and Razgon, is called the shadow removal technique and is based on a random sampling of important separators. We explain both techniques and illustrate them on the so-called MULTIWAY CUT problem. These methods lead to the most interesting parameterized algorithms for graph separation problems in the last few years.

One of the goals of my PhD was to find upper bounds on τ . One of the best tools for obtaining such bounds is the VC-dimension. The whole Chapter 3 is devoted to its study. This chapter is built in order to give examples, intuitions and theorems on this notion. We also mention many applications of these results in graph theory. The key lemma of the VC-dimension theory, due to Haussler and Welzl, ensures that the gap between τ and τ^* is bounded whenever the VC-dimension is bounded. After sketching its proof we will give several of its applications to graph theory.

Unfortunately, the class of hypergraph of bounded VC-dimension does not have the Erdős-Pósa property. Nevertheless, we give two generalizations of the result of Haussler and Welzl which ensure that $\tau \leq f(\nu)$. First we introduce the notion of 2VC-dimension which strengthens the notion of VC-dimension. Ding, Seymour Winkler proved that a bounded 2VC-dimension ensures the Erdős-Pósa property. In the second generalization, due to Matoušek, we need notions of both VC-dimension and (p, q) -property in order to bound the gap between τ and ν .

All along Chapter 3, we give applications to graph theory. One of them, due to Stéphan Thomassé and myself, is a partial result on a graph coloring conjecture of Scott (the general conjecture has been disproved since). More precisely we proved that the chromatic number of any maximal triangle-free graph with no induced subdivision of a graph H can be bounded by a function of $|H|$. The complete proof is in Section 3.4.2.

One of the first applications of the VC-dimension in graph theory is due to Chepoi, Estellon and Vaxès. They proved that there exists a constant c such that every planar graph of diameter 2ℓ can be covered by c balls of radius ℓ (where c does not depend on ℓ). Their proof is based on a hypergraph argument since they consider the hypergraph of balls of radius ℓ . Using VC-dimension they proved that the transversality of this hypergraph is bounded which leads to their result. In Chapter 4 we present a generalization of their result due to Stéphan Thomassé and myself. We first propose a formal definition of VC-dimension and 2VC-dimension for graphs. The interest of this notion on graphs is twofold. First we prove that any class of graphs with no large clique-minor, or with no large clique-width, has bounded 2VC-dimension. So this notion catches both notions of minors

and clique-width.

Then we extend the result of Chepoi, Estellon and Vaxès to graphs of bounded 2VC-dimension (which is a direct consequence of a theorem of Chapter 3) and then to graphs of bounded VC-dimension (using the proof scheme of the paper of Chepoi, Estellon and Vaxès with more involved arguments). More precisely we prove that for every r , the minimum number of balls of radius r needed to cover the graph can be bounded by a function of the maximum packing of balls of radius r and of the VC-dimension of the graph. For the more special class of graphs of bounded 2VC-dimension the upper bound is smaller and the proof is shorter.

In Chapter 5 we study with Aurélie Lagoutte and Stéphan Thomassé a conjecture of Yannakakis via VC-dimension. The conjecture states the following: given a graph G , there exists a polynomial number (in the size of G) of bipartitions B of the vertex set such that for any clique K and any stable set S which do not intersect, there exists a bipartition of B such that all the vertices of K are in one part and all the vertices of S are in the other part. We prove that this conjecture holds for several classes of graphs such as random graphs, split-free graphs and graphs with neither long paths nor their complements. For split-free graphs, the proof is based on a VC-dimensional argument on a particular neighborhood hypergraph. For graphs with neither long paths nor their complements, the proof is a consequence of another result due to Aurélie Lagoutte, Stéphan Thomassé and myself on the Erdős-Hajnal conjecture. More precisely we prove that the so-called Erdős-Hajnal conjecture holds for graphs with neither long paths nor their complements. The Yannakakis' conjecture on this class of graphs is obtained as a corollary of this result.

We then study more specifically the links between this conjecture and another conjecture of graph theory called the Alon-Saks-Seymour conjecture. The last conjecture states that if we consider a graph G whose edges can be covered by k edge-disjoint complete bipartite graphs then the chromatic number of G is at most $P(k)$ where P is a fixed polynomial function. We prove that the Yannakakis' conjecture is equivalent to the Alon, Saks, Seymour conjecture. In addition we show that these conjectures are linked with some Constraint Satisfaction Problems, and in particular with the so-called Stubborn problem which was recently proved to be polynomial.

A last important type of hitting sets problems studied in this manuscript is the graph separation problems. In Chapter 6, we study the MULTICUT problem from an parameterized algorithmic point of view. Let G be a graph and R be a set of pairs of vertices. A multicut of (G, R) is a set of edges whose deletion disconnect every pair of vertices of R (*i.e.* for every $(x, y) \in R$, x and y are not in the same connected component). In other words a multicut is a hitting set of the set of paths between vertices of R .

MULTICUT:

Input: A graph $G = (V, E)$, a set of pairs of vertices R , an integer k .

Parameter: k .

Output: TRUE if there is a multicut of (G, R) of size at most k , FALSE otherwise.

We prove with Jean Daligault and Stéphan Thomassé that MULTICUT is FPT parameterized by the size of the solution, *i.e.* there exists an algorithm running in $f(k) \cdot n^c$ deciding the MULTICUT problem. It was considered as one of the main open problems of parameterized complexity. At the same time, Marx and Razgon proposed another proof of the same result. Their proof is based on the shadow removal technique introduced in Chapter 2. Our proof is based on the important sep-

arators (introduced in Chapter 2) but also on other combinatorial techniques such as Δ -systems, Dilworth's theorem and iterative compression.

Publications

Throughout this manuscript, several of my works are presented. However, for the coherence of this manuscript, some of them were omitted. All the results obtained during my PhD are listed below.

International conferences:

- [34] *A Polynomial Kernel for Multicut in Trees* with Jean Daligault, Stéphan Thomassé, Anders Yeo, **STACS'09**, (2009) 183-194. In this paper, we proved that MULTICUT admits a polynomial kernel when the input graph is a tree. More precisely we proved that there exists a kernel of size $\mathcal{O}(k^6)$. It means that we can reduce in polynomial time any instance into an instance of size $\mathcal{O}(k^6)$ such that the reduced instance is positive if and only if the initial one is positive¹. This kernel was improved into a cubic kernel in [51].
- [39] *Equivalence and Inclusion Problem for Strongly Unambiguous Büchi Automata* with Christof Löding, **LATA'10**, (2010) volume 6031 of Lecture Notes in Computer Science 118-129. Unambiguous automata are automata such that any accepted word has exactly one accepting sequence. The equivalence and the inclusion problems are polynomial time decidable for finite unambiguous automata instead of PSPACE-complete for general automata. We generalized this result for strongly unambiguous Büchi automata (automata on infinite words)².
- [33] *Multicut is FPT* with Jean Daligault, Stéphan Thomassé, **STOC'11**, Proceedings of the 43rd annual ACM symposium on Theory of computing (2011), 459-468. We proved that MULTICUT admits an FPT algorithm parameterized by the size of the solution. The proof of this result will be presented in Chapter 6.
- [35] *Parameterized Domination in Circle Graphs* with Daniel Gonçalves, George Mertzios, Christophe Paul, Ignasi Sau and Stéphan Thomassé, **WG'12**, volume 7551 of Lecture Notes in Computer Science 308-319 (2012). A circle graph is an intersection graph of chords of a circle. In this article we proved that DOMINATING SET and lots of its variants are $W[1]$ -hard (*i.e.* probably do not admit FPT algorithms): namely DOMINATING SET, CONNECTED DOMINATING SET, INDEPENDENT DOMINATING SET and TOTAL DOMINATING SET are $W[1]$ -hard restricted to circle graphs. We also provided a polynomial time algorithm for TREE DOMINATING SET.
- [29] *Recoloring bounded treewidth graphs* with Marthe Bonamy, **LAGOS'13** (2013). Given two proper colorings of a graph, one can ask if it is possible to transform one coloring into the other by recoloring one vertex at each step, and such that at each step the current coloring is proper. We proved that any graph can be recolored with a quadratic number of steps if the number of colors is at least the treewidth of the graph plus 2.

1. This result was obtained during my bachelor internship.

2. This result was obtained during my first year of Master internship.

- [30] *Adjacent vertex-distinguishing edge coloring of graphs* with Marthe Bonamy and Hervé Hocquard, accepted to **EuroComb'13**. An AVD-coloring is a proper edge-coloring such that any pair of adjacent vertices is not adjacent to the same set of colors. A conjecture states that any graph on at least six vertices is AVD $(\Delta + 2)$ -colorable (where Δ denotes the maximum degree). We gave some evidence for this conjecture by proving stronger results on graphs of bounded maximum average degree and planar graphs with $\Delta \geq 12$.

Journals.

- [41] *Scott's induced subdivision conjecture for maximal triangle-free graphs* with Stéphan Thomassé, **Combinatorics, Probability and Computing**, (2012) 21:512-514. A proof of this result is presented in Chapter 3.
- [36] *Parameterized Domination in Circle Graphs* with Daniel Gonçalves, George Mertzios, Christophe Paul, Ignasi Sau and Stéphan Thomassé, **Theory of Computing Systems** (2013) 1-28. This paper is the long version of [35].

Submitted papers.

- [37] *Clique versus independent set*, with Aurélie Lagoutte and Stéphan Thomassé. A complete version of this article will be found in Chapter 5.
- [38] *The Erdős-Hajnal Conjecture for Paths and Antipaths*, with Aurélie Lagoutte and Stéphan Thomassé. The Erdős-Hajnal conjecture states the following: every graph on n vertices with no induced copy of a fixed graph H contains a clique or an induced stable set of size n^ϵ where ϵ is a strictly positive constant which only depends on H . This conjecture is open even for $H = P_5$. We proved in this note that any graph which contains (as an induced copy) neither the graph P_k or its complement, admits a clique or a stable set of size at least $n^{\epsilon(k)}$. The proof of this result is presented in Chapter 5.
- [40] *VC-dimension and Erdős-Pósa property of graphs*, with Stéphan Thomassé. This result is presented in Chapter 4.
- [1] *Excluding cycles with a fixed number of chords*, with Pierre Aboulker. Trotignon and Vušković proved that the class of graphs with no cycle with exactly one chord is χ -bounded (*i.e.* the chromatic number can be bounded by a function of the maximal clique). We generalized their result for graphs with no cycle with exactly two chords and with no cycle with exactly three chords. More precisely, we proved that in both cases we have $\chi \leq \omega + c$ where c is a constant.
- [20] *Rainbow colorings of 3-chromatic graphs*, with Stéphane Bessy. A rainbow coloring of G is a coloring of G with $\chi(G)$ colors such that each vertex is the beginning of a path of length $\chi - 1$ containing a vertex of each color. It is conjectured that every graph except C_7 has a rainbow coloring. We proved that every 3-chromatic graph except C_7 admits a rainbow coloring. We also gave some evidence for the case of 4-chromatic graphs.
- [32] *Sparsest problem on chordal graphs*, with Marin Bougeret, Rodolphe Giroudeau and Rémi Watrigant. The SPARSEST problem consists in, given a graph G and an integer k , finding k vertices of G inducing the fewest number of edges. It generalizes the INDEPENDENT SET problem: if there is an independent set of size k then the sparsest set on k vertices induces no edge. We

proved that this problem is FPT and does not admit a polynomial kernel restricted to chordal graphs and parameterized by k .

Preprints.

- [28] *Recoloring graphs via tree-decompositions*, with Marthe Bonamy. This paper is a long version of [29]. In addition we gave some bounds on the number of steps needed to recolor cographs and distance-hereditary graphs. More precisely we proved that cographs can be recolored with a linear number of steps and distance-hereditary graphs with a quadratic number of steps as long as the number of colors is at least the chromatic number plus one.

Preliminaries

In this chapter, we introduce the main definitions concerning graphs and hypergraphs and parameterized complexity which will be used all along the manuscript.

In this thesis, we only deal with finite structure. So in all the manuscript, we consider a finite set V called the *vertex set*; the elements of V are called the *vertices* and its cardinality $|V|$ is denoted by n .

vertex set
vertices

Sets and orders. Let V be a set of vertices and X be a subset of V . The *complement of X* is the set $V \setminus X$ and we denote it by \overline{X} . A (*partial*) *order* $<$ is a transitive and anti-symmetric relation. *Transitive* means that $x < y$ and $y < z$ implies $x < z$. And *anti-symmetric* means that $x < y$ implies $y \not< x$. Two elements x and y are *comparable* if $x < y$ or $y < x$. A *total order* is an order such that any pair of elements is comparable. An *antichain* of a partial order is a subset of pairwise incomparable elements. A *chain* is a subset of pairwise comparable elements. Note that the order restricted to the elements of a chain is a total order. A set of chains *covers* V if every vertex of V appears in at least one chain. The well-known Dilworth's theorem links the size of coverings and the size of antichains.

\overline{X}
partial
order
comparable
antichain
chain
covers V

Theorem 1.1 (Dilworth 1950). *For every order over a finite set, the maximum size of an antichain equals the minimum number of chains needed to cover the whole set V .*

If there is an antichain of size at least k , then the number of chains needed to cover V is at least k . Indeed elements of an antichain are pairwise incomparable, so no two of them can be in the same chain. Dilworth's theorem ensures that the reverse inequality also holds.

1.1 Hypergraphs

For more definitions and results concerning hypergraphs, the reader is referred to [19]. Let V be a set of vertices. Let E be a subset of all subsets of V . The elements of E are called (*hyper*)*edges* and the whole set E is called the (*hyper*)*edge set*. In the following and all along the manuscript, we will consider that the empty set can be a hyperedge. The *size* of a hyperedge is its number of vertices. A

hyperedge

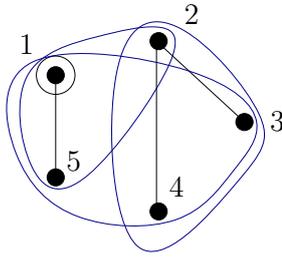


Figure 1.1: A hypergraph. For readability the hyperedges of size at least 3 are represented in blue.

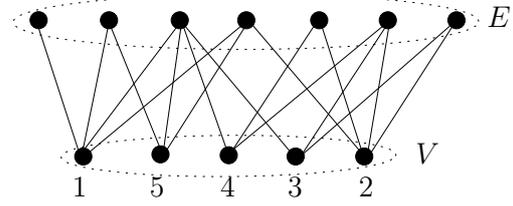


Figure 1.2: Incidence bipartite graph of Figure 1.1.

hypergraph *hypergraph* H is a pair (V, E) where V is a set of vertices and E is a set of hyperedges. In the following, H will always denote a hypergraph, V its vertex set and E its hyperedge set. The cardinality of E , denoted by m , is the number of hyperedges of H . Figure 1.1 represents a hypergraph with hyperedge set $a = \{1\}$, $b = \{1, 5\}$, $c = \{1, 3, 4, 5\}$, $d = \{1, 2, 5\}$, $e = \{2, 4\}$, $f = \{2, 3, 4\}$ and $g = \{2, 3\}$. Straight lines are hyperedges of size 2 (in Figure 1.1 and throughout this manuscript).

simple All the invariants considered during this manuscript are not modified if the same edge appears several times. So, when we count the number of hyperedges, we will assume that the hypergraph is *simple*, *i.e.* does not contain two identical hyperedges. The number of hyperedges of a (simple) hypergraph is at most 2^n since a set of size n contains 2^n distinct subsets.

trace Let H be a hypergraph. Let V' be a subset of vertices. The *trace* of a hyperedge e on V' is $e \cap V'$. The *restriction of H to V'* , denoted by $H[V']$ is the hypergraph on vertex set V' where hyperedges are the traces of the hyperedges of H on V' . A hypergraph H' is a *subhypergraph* of H if H' can be obtained from H by a sequence of deletions of vertices and hyperedges. In other words, a hypergraph H' is a subhypergraph of H if there is a subset of vertices V' such that H' can be obtained from $H[V']$ by hyperedges deletion.

graph **Incidence bipartite graph.** A *graph* is a hypergraph with hyperedges of size exactly 2. Let $H = (V, E)$ be a hypergraph. The *incidence bipartite graph* G of H has vertex set (V, E) and ve is an (oriented) edge of G if and only if $v \in e$ in H . Note that the hypergraph can be reconstructed from its incidence bipartite graph. Figure 1.2 is the incidence bipartite graph of Figure 1.1. Note that there is an orientation on the edges (from V to E) in order to keep in mind which set represents the vertices of the hypergraph and which set represents the hyperedges of the hypergraph.

dual hypergraph **Dual hypergraph.** The *dual hypergraph* of H , denoted by H^d , is the hypergraph on vertex set E where for every $v \in V$ there exists a hyperedge e_v in H^d such that the vertex e of H^d is in e_v if and only if $v \in e$ in H . Informally, it means that every hyperedge e becomes a vertex in H^d and every vertex v becomes a hyperedge e_v in H^d . The hyperedge e_v in H^d contains all the $e \in E(H)$ such that $v \in e$ in the hypergraph H . Note that it corresponds to an exchange of the roles of V and E in the incidence bipartite graph; instead of considering that the edges are from V to E we consider that the edges are from E to V . Figure 1.3 represents the dual of the hypergraph of Figure 1.1.

Observation 1.2. Every hypergraph H satisfies $(H^d)^d = H$.

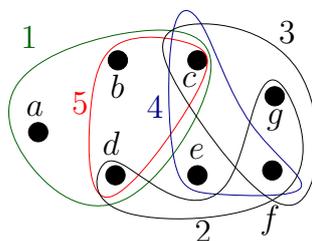


Figure 1.3: Dual hypergraph of Figure 1.1. In order to understand the construction, the hyperedges of Figure 1.1 are called a, b, c, d, e, f, g in their order of apparition (from left to right) in the incidence bipartite graph of Figure 1.2.

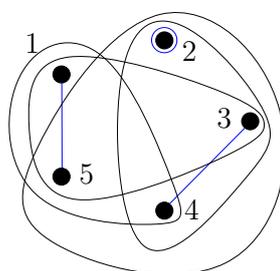


Figure 1.4: The complement hypergraph of Figure 1.1. Complements of blue hyperedges are represented in blue.

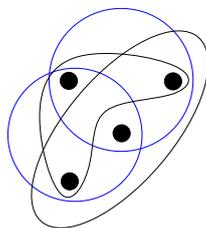
Proof. Let $H = (V, E)$ be a hypergraph and let $((V, E), E')$ be its incidence bipartite graph. By definition, the incidence bipartite graph of H^d is $((E, V), E')$. And the incidence bipartite graph of $(H^d)^d$ is $((V, E), E')$, i.e. $(H^d)^d = H$. □

A hypergraph is *auto-dual* if $H^d = H$. All along the manuscript, we will introduce several classes of auto-dual hypergraphs. auto-dual

Opposite and complement hypergraphs. The *complement hypergraph* H^c of $H = (V, E)$ is the hypergraph on vertex set V where X is a hyperedge if and only if $V \setminus X \in E$. In other words, the complement hypergraph is obtained by replacing every hyperedge by its complement. Figure 1.4 represents the complement hypergraph of Figure 1.1. complement

The *opposite hypergraph* H^o of H has vertex set V and e is a hyperedge of H^o if and only if e is not a hyperedge of H . So any set that is not a hyperedge of H becomes a hyperedge in the opposite hypergraph. The opposite hypergraph of Figure 1.1 contains $2^5 - 7 = 25$ hyperedges. opposite hypergraph

Uniform and complete hypergraphs. Let k, n be two integers with $k \leq n$. A hypergraph is *k-uniform* if all its hyperedges have size exactly k . Note that a k -uniform hypergraph has at most $\binom{n}{k}$ (distinct) hyperedges. The *complete k-uniform hypergraph* on n vertices, denoted by $\mathcal{U}_{k,n}$ is the k -uniform hypergraph on n vertices where every subset of size k is a hyperedge. Note that the complete uniform hypergraph $\mathcal{U}_{k,n}$ has exactly $\binom{n}{k}$ hyperedges. The complete uniform hypergraph k-uniform $\mathcal{U}_{k,n}$

Figure 1.5: Representation of $\mathcal{U}_{3,4}$.

$\mathcal{U}_{3,4}$ is represented in Figure 1.5 and $\mathcal{U}_{2,4}$ in Figure 1.6(a).

The k -complete hypergraph on n vertices, denoted by $\mathcal{C}_{k,n}$ is the hypergraph on n vertices where every subset of size at most k is a hyperedge. In other words, the k -complete hypergraph is the (edge) union of the $\mathcal{U}_{k',n}$ with $k' \leq k$. The complete hypergraph on n vertices, denoted by \mathcal{C}_n , is the hypergraph $\mathcal{C}_{n,n}$, i.e. the hypergraph containing all the possible hyperedges.

1.2 Graphs

For complements on graph theory, the reader is referred to [19, 31, 70]. Recall that graphs are 2-uniform hypergraphs. In the following $G = (V, E)$ denotes a graph. In order to simplify notations, the edge $\{u, v\}$ is denoted by uv . As for hypergraphs, we denote $|V|$ by n and $|E|$ by m . Two vertices u and v are *adjacent* (or *connected*) if $uv \in E$. We also say that u is a *neighbor* of v . The vertices u and v are the *endpoints* of the edge uv . The (*open*) *neighborhood* of a vertex u , denoted by $N_G(u)$ (or $N(u)$ when no confusion is possible) is the subset of vertices v of V such that uv is an edge. For every subset X of V , $N(X)$ will denote the set $\bigcup_{x \in X} N(x)$. The *degree* of u is its number of neighbors. The set $N(u) \cup \{u\}$, denoted by $N[u]$, is the *closed neighborhood* of u . The *closed non-neighborhood* $N^C[x]$ of x is $V \setminus N[x]$. We define similarly the non-neighborhood of x which is $V \setminus N(x)$ by $N^C(x)$.

Operation on graphs. The *complement* of G is the graph $G^c = (V, E^c)$ where $uv \in E^c$ if and only if $uv \notin E$. It is coherent with the hypergraph definition of complement if we restrict to hyperedges of size 2. Let V_1 be a subset of vertices. The (*sub*)*graph induced by* V_1 , denoted by $G[V_1]$, has vertex set V_1 and its edges are the edges of G with both endpoints in V_1 . More formally $G[V_1] = (V_1, E')$ where $xy \in E'$ if and only if $x, y \in V_1$ and $xy \in E$. A graph G contains an *induced copy* of F if there exists $V_1 \subseteq V$ such that $G[V_1]$ is (isomorphic to) F . A graph G *contains a copy of* F (or F is a *subgraph of* G) if there exists $V_1 \subseteq V$ such that F can be obtained from $G[V_1]$ by a deletion of edges. Informally it means that $G[V_1]$ “contains” the edges of F , i.e. is a “super-graph” of F (for the containment). Note that a graph containing an induced copy of F also contains a copy of F but the reverse is not true (the graph Figure 1.6(a) contains Figure 1.6(b) as a copy but not as an induced copy). A graph is *F-free* if it does not contain any induced copy of F .

Graph classes. A *class of graphs* (or a *family of graphs*) is a (non-necessarily finite) set of graphs. A class \mathcal{C} is *hereditary* if every induced subgraph of a graph of \mathcal{C} is in \mathcal{C} . Let \mathcal{H} be a class of graphs.

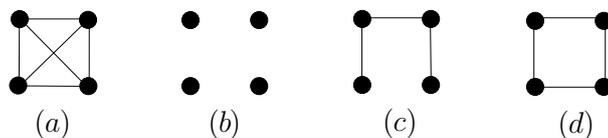


Figure 1.6: (a) A clique of size 4. (b) A stable set of size 4. (c) A path of length 3. (d) A cycle of length 4.

A class of graphs \mathcal{C} is \mathcal{H} -free if for every $H \in \mathcal{H}$ and $G \in \mathcal{C}$ the graph G is H -free.

Clique and Stable Sets. A *clique* (or a *complete graph*) K of a graph G is a subset of pairwise adjacent vertices of G . In other words, it is a subset K such that $G[K]$ contains all the possible edges. The *clique number* of G , denoted by $\omega(G)$, ω is the size of a maximum clique in G . The clique of size 3 is often called a *triangle*. The clique of size n is denoted by K_n and K_4 is represented in Figure 1.6(a). The clique of size n is exactly the hypergraph $\mathcal{U}_{2,n}$. By abuse of notation, a clique will denote both the subset of vertices which are pairwise connected and the graph induced by these vertices. Note that, unlike general graphs, a graph contains a copy of a clique K if and only if it contains an induced copy of a clique K . A *stable set* (or an *independent set*) S is a subset of vertices which are pairwise non-adjacent. In other words, it is a subset S such that $G[S]$ contains no edge. The *independence number*, denoted by $\alpha(G)$ α is the size of a maximum stable set. The stable set of size 4 is represented in Figure 1.6(b). Note that any stable set (resp. clique) of G^c is a clique (resp. stable set) of G . Remark also that a clique and a stable set intersect on at most one vertex.

clique
 ω
triangle
stable set
 α

Random graphs. Let n be a positive integer and $p \in [0, 1]$. The random graphs considered in this manuscript are drawn under the Erdős-Rényi model. The random graph $G(n, p)$ is a probability space over the set of graphs on the vertex set $\{1, \dots, n\}$ determined by $\Pr[ij \in E] = p$, where these events are mutually independent. We say that $G(n, p)$ has clique number ω if ω is the minimum integer that satisfies $\mathbb{E}(\text{number of cliques of size } \omega) \geq 1$. We define similarly the independence number α of $G(n, p)$. An event \mathcal{E} occurs *with high probability* if the probability of this event tends to 1 when n tends to infinity.

Ramsey's type theorems. A *k-edge coloring* γ of a clique K if a function $\gamma : E(K) \rightarrow \{1, \dots, k\}$. A *monochromatic clique* of an edge coloring γ of K is a subset of vertices S such that all the edges of $K[S]$ are colored identically.

k-edge coloring

Theorem 1.3 (Ramsey). *For every k , there exists a function R_k such that any k -edge coloring of $K_{R_k(n)}$ has a monochromatic clique of size n .*

Proof. Let γ be an edge-coloring of a clique. The coloring γ is a *right coloring* on the ordered set v_1, \dots, v_n of vertices if for every q, r, s with $r, s > q$ we have $\gamma(v_q v_r) = \gamma(v_q v_s)$. Informally, it means that all the increasing edges leaving a same vertex are colored identically. Note that it does not necessarily mean that all the edges adjacent to the same vertex are colored identically since the "decreasing" edges can be colored differently.

Let us prove by induction that there exists a function Q_k such that every graph of size at least $Q_k(n)$ has a subset of n ordered vertices inducing a right coloring for γ . Let us prove that $Q_k(n + 1) \leq$

$k \cdot Q_k(n) + 1$. Let K be a clique of size $k \cdot Q_k(n) + 1$ and γ be a k -edge coloring of K . Let $u \in V$. For every $i \in \{1, \dots, k\}$, let V_i be the set of vertices v of K such that uv is colored with i . Note that $(V_i)_{i \leq k}$ is a partition of $V \setminus \{u\}$ (since K is a clique). Up to a permutation on the color set, we can assume that V_1 has maximum cardinality over the sets $\{V_1, \dots, V_k\}$. Since $V \setminus \{u\}$ is partitioned into at most k sets, the size of V_1 is at least $Q_k(n)$.

By induction hypothesis, the clique $K[V_1]$ contains an ordered subset U of size n such that γ is a right coloring on U . Let us denote by u_1, \dots, u_n the ordered set of vertices of U . Let us prove that γ is a right coloring on $U' = \{u, u_1, \dots, u_n\}$. By induction hypothesis, γ is a right coloring on U . So γ is a right coloring on U' if and only if all the edges uu_i are colored identically, which holds since, by construction, all the vertices of U are in V_1 .

Let K be a clique of size $Q_k(kn)$. The previous part of the proof ensures that K contains a subset W of size kn inducing a right coloring for γ . Since there are at most k colors, at least n vertices have the same color "at their right". Denote by W' this subset. The set $K[W']$ is a monochromatic clique of size n . So $R_k(n) \leq Q_k(kn)$. \square

Remark that since Theorem 1.3 holds for cliques of size $R_k(n)$, then it also holds for any larger clique. It can be derived from the proof of Theorem 1.3 that the function R_k given by the proof of Theorem 1.3 satisfies $R_k(n) \approx k^{kn}$. In other words, any k -edge coloring of a clique of size n admits a monochromatic clique of size almost $(\log_k n)/k^1$. Theorem 1.3 is much more famous over the following form:

Corollary 1.4. *There exists a function R such that every graph G on at least $R(n)$ vertices has an (induced) clique or a stable set of size n .*

Proof. Let $G = (V, E)$ be a graph. Let γ be the 2-edge coloring of the clique K of size $|V|$ where edges of G are colored with 1 and non-edges with 2. A monochromatic clique of color 1 induces a clique in G and a monochromatic clique of color 2 induces a stable set. If $|V| \geq R(n)$ then K contains a monochromatic clique of size n and then G contains a clique or a stable set of size n . So there is no monochromatic clique of size n in γ and then Theorem 1.3 ensures that $|V(G)| \leq R_2(n)$. \square

The function R of Corollary 1.4 almost satisfies $R(n) \approx 4^n$. On the opposite, random graph $G_{n,1/2}$ has (with high probability) no clique of size larger than $2 \log n$. In other words, we know that $R(n) \geq \sqrt{2}^n$. So we know that the Ramsey number $R(n)$ satisfies:

$$\sqrt{2}^n \leq R(n) \leq 4^n.$$

Closing this gap is a widely open problem. Nevertheless, for several classes of graphs, the size of maximum cliques and of maximum stable sets is unbalanced. For instance, the size of a maximum clique of a K_ℓ -free graph is at most $\ell - 1$. And in this case, the size of a stable set is much more larger than $\log n$. A *triangle-free* graph is a K_3 -free graph.

Observation 1.5. *Every triangle-free graph on n vertices has a stable set of size at least \sqrt{n} .*

1. All along this manuscript \log_b denotes the logarithm to base b and \log denotes the logarithm to base b

Sketch of the proof. First assume that a vertex u has degree at least \sqrt{n} , then denote by X the neighborhood of u . The set X is a stable set. Indeed if xy is an edge of $G[X]$ then $G[\{u, x, y\}]$ would be a triangle, a contradiction. Thus X is a stable set of G of size \sqrt{n} .

So we can assume that every vertex has degree less than \sqrt{n} . Consider the following algorithm: as long as there remain vertices, add a vertex x in the solution and delete the neighborhood of x from G (i.e. replace G by $G[V \setminus N[x]]$). This algorithm provides a stable set of size \sqrt{n} . Indeed, at each step at most \sqrt{n} vertices are deleted and one vertex is added in the solution. Since at each step, we delete the neighborhood of the chosen vertex from the vertex set, the set of chosen vertices is a stable set. \square

This square root lower bound has been improved by Kim who proved the following in [127].

Theorem 1.6 (Kim [127]). *Every triangle-free graph G on n vertices admits a stable set of size $\Omega(\sqrt{n} \cdot \log n)$.*²

Paths and distances. Given a graph $G = (V, E)$, a *walk* of length k from $x \in V$ to $y \in V$ is a sequence of vertices $x = x_0, x_1, \dots, x_{k-1}, x_k = y$ where $x_i x_{i+1} \in E$ for each $0 \leq i \leq k-1$. A *path* is a walk with pairwise distinct vertices. Note that a path is not necessarily induced. The vertices x and y are the *endpoints* of the walk. The $x_i x_j$ -*subpath* is the path x_i, x_{i+1}, \dots, x_j . The *interior* of the path is the $x_1 x_{k-1}$ -subpath. The *length* of a path is its number of vertices minus one. The path of length $(k-1)$ is denoted by P_k and P_4 is represented in Figure 1.6(c). A graph G contains a P_k if a subgraph of G is isomorphic to P_k .

walk
path
interior
length

A *cycle* is a path u_1, \dots, u_k such that $u_k u_1$ is also an edge and such that $k \geq 3$. The *length* of a cycle is its number of vertices. The induced cycle of length k is denoted by C_k . The cycle C_4 is represented in Figure 1.6(d). A graph G contains a C_k if a subgraph of G is isomorphic to C_k . The *girth* of a graph G is the minimum size of a cycle of G . Any cycle of the length of the girth is necessarily induced as otherwise there would be a (strictly) shorter cycle. Also note that a graph is triangle-free if and only if its girth is at least 4.

cycle
girth

A *minimum path* from x to y , also called *minimum xy -path*, is a path of minimum length from x to y . The *distance between x and y* , denoted by $d(x, y)$ is the length of a minimum path from x to y when such a path exists and $+\infty$ otherwise. The distance between a set X and a set Y is the minimum of the distances between x and y for all pairs $(x, y) \in X \times Y$. The *ball of center x and radius k* , denoted by $B(x, k)$, is the set of vertices at distance at most k from x . Note that the vertices of $B(x, 1)$ are the vertices of the closed neighborhood of x .

minimum
path

Chromatic number. A vertex-coloring (or *coloring* for short) is a function $\gamma : V \rightarrow \{1, \dots, k\}$. A *proper k -coloring* is a coloring such that every pair of adjacent vertices receives distinct colors. The minimum k for which there exists a proper k -coloring is called the *chromatic number* of the graph and is denoted by $\chi(G)$. Note that $\omega(G) \leq \chi(G)$ since, for every clique K of G , the vertices of K must receive pairwise distinct colors (since the vertices are adjacent). Lots of constructions ensure that the gap between the chromatic number and the clique number can be arbitrarily large (see [85, 155, 165, 194] for instance).

coloring
chromatic
number
 χ

2. By $\Omega(n)$, we mean at least cn for some positive constant c .

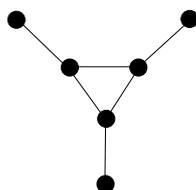


Figure 1.7: A net: a graph made of a triangle and three pending edges. A net is a split graph.

Observation 1.7. *The chromatic number of a graph equals the minimum number of stable sets which cover its vertices.*

Proof. Let G be a graph and γ be a (proper) $\chi(G)$ -coloring of G . For every color i , the graph induced by the vertices of color i is a stable set. Since G is properly colored with at most k colors, V is covered by the union of at most k stable sets. Conversely, if k stable sets cover V , then coloring the vertices of the same stable set with the same color provides a proper coloring. \square

The *stable set hypergraph* of a graph G is the hypergraph H where vertices are maximal (by inclusion) stable sets of G , and for every vertex v we create a hyperedge e_v containing all the maximal stable sets containing v . In other words, for every vertex $v \in V$, there is a hyperedge e_v such that e_v contains all the maximal stable set S of $V(H)$ such that $v \in S$.

Bipartite. A graph is *bipartite* if its vertex set can be partitioned into two sets V_1, V_2 such that every edge has one endpoint in V_1 and one in V_2 , *i.e.* neither $G[V_1]$ nor $G[V_2]$ contains an edge. Bipartite graphs are denoted by $((V_1, V_2), E)$ where $V_1 \cup V_2$ is the vertex set and E is the set of edges. Figure 1.2 is a bipartite graph. Note that the chromatic number of bipartite graphs equals 2 (as long as they contain at least one edge). Indeed, V can be partitioned into two stable sets V_1 and V_2 , so Observation 1.7 ensures that $\chi(G) \leq 2$. Two subsets of vertices $X, Y \subseteq V$ are *completely adjacent* if for all $x \in X, y \in Y, xy \in E$. They are *completely non-adjacent* if there is no edge between them. A bipartite graph on vertex set (V_1, V_2) which are completely adjacent is a *complete bipartite graph* and is denoted by $K_{|V_1|, |V_2|}$. The graph $K_{2,2}$ is represented in Figure 1.6(d).

Split graphs. A graph $G = (V, E)$ is *split* if $V = V_1 \cup V_2$ and the subgraph induced by V_1 is a clique and the subgraph induced by V_2 is a stable set. Figure 3.9 represents a split graph, called the net.

Trees and connectivity. A graph G which does not contain any cycle is *acyclic* or is a *forest*. It is classical to show that any forest with at least one edge contains at least two vertices of degree one. The *leaves* of a forest are the vertices of degree at most one. The vertices of a forest are often called *nodes*. A graph is *connected* if for every pair of vertices u and v , there exists a path from u to v . In other words, a graph is connected if for every pair of vertices the distance is not infinite. A *connected component* of a graph is a maximum subset of vertices which induces a connected graph. A *tree* is a connected acyclic graph. Figure 1.6(b) and (c) represent forests but only Figure 1.6(c) is a tree. One can easily verify that every tree on n vertices has exactly $n - 1$ edges. It is classical to show that a graph is connected if it is a spanning tree.

feedback
vertex set

A *feedback vertex set* X of a graph G is a subset of vertices whose deletion provide a graph without cycle, *i.e.* such that $G[V \setminus X]$ is acyclic.

Dominating set. Given a graph $G = (V, E)$, a *dominating set* of G is a subset of vertices X such that for every vertex $v \in V \setminus X$, there exists a vertex x of X such that xv is an edge. In other words a dominating set is a subset X of vertices such that $N[X] = V$. Several variants of dominating sets exist, such as total, independent, connected dominating sets. In this manuscript in particular, we are interested by dominating sets at large distances. A vertex u *dominates a vertex v at distance k* if there exists a path of length at most k with endpoints u and v . A dominating set is a subset of vertices which dominates every vertex at distance 1.

dominating
set

The *neighborhood hypergraph* of $G = (V, E)$ is the hypergraph on vertex set V where $e \subseteq V$ is a hyperedge if and only if there exists a vertex $v \in V$ such that $N(v) = e$ in G . Informally, hyperedges are open neighborhoods of vertices of the graph. One can note that a subset of vertices of the hypergraph intersecting all the hyperedges is a dominating set of the graph G . One can similarly define the closed neighborhood hypergraph where hyperedges are the closed neighborhoods of the vertices of G instead of their open neighborhoods.

The *B -hypergraph* of a graph G has vertex set V and a subset $Y \subseteq V$ is a hyperedge if there are a vertex $x \in V$ and an integer k such that $Y = B(x, k)$. For a given integer ℓ , the *B_ℓ -hypergraph* of G has vertex set V and $Y \subseteq V$ is a hyperedge if and only if there is an x such that $Y = B(x, \ell)$. Recall that the closed neighborhood hypergraph is the B_1 -hypergraph. Note that, for every ℓ the B_ℓ -hypergraph of a graph G and its dual are the same since for every pair x, y of vertices, $x \in B(y, \ell)$ if and only if $y \in B(x, \ell)$.

B -
hypergraph
 B_ℓ -
hypergraph

Directed graphs. For more details concerning directed graphs, the reader is referred to [18]. An *arc* is an oriented pair of vertices. For brevity, we will denote the ordered pair (u, v) by uv when no confusion is possible with edge of graphs. Nevertheless keep in mind that uv is distinct from vu in directed graphs while it refers to the same edge in undirected graphs. A *directed graph* (or *digraph* for short) is a pair $D = (V, A)$ such that V is a set of vertices and A is a set of arcs. For every arc uv , u is called the *beginning of uv* and v the *end of uv* . For every vertex u , the *in-neighborhood* (resp. *out-neighborhood*) of u , denoted by $N^-(u)$ (resp. $N^+(u)$) is the set of vertices v such that vu (resp. uv) is an arc. The *closed in-neighborhood of u* is the in-neighborhood of u plus the vertex u itself. The *closed in-neighborhood hypergraph H* of a directed graph D is the hypergraph on vertex V where $X \subseteq V$ is a hyperedge if and only if X is the closed in-neighborhood of some vertex v in D .

arc

directed
graph

A directed graph is *oriented* if for every pair of vertices both uv and vu are not arcs. An oriented graph is an orientation of the edges of a simple graph. The *underlying graph* of an oriented graph is the (unoriented) graph such that xy is an edge if and only if xy or yx are arcs. In other words, the underlying graph is the graph obtained by “forgetting” the orientations of the arcs. A directed path is a subset of vertices u_1, \dots, u_ℓ such that $u_i u_{i+1}$ is an arc for every $1 \leq i \leq \ell - 1$. A *circuit* of a directed graph is a directed path u_1, \dots, u_ℓ such that $u_\ell u_1$ is an arc. Note that a directed graph is oriented if it does not contain any circuit of length 2. An oriented graph is a *tournament* if for every pair u, v of vertices either uv or vu are arcs. In other words, a tournament of size n is an orientation of the clique K_n . A *transitive tournament* is an acyclic tournament, *i.e.* a tournament with no circuit.

oriented
underlying
graph

circuit

tournament

transitive
tournament

Oriented graphs and orders. An oriented graph is transitive whenever for any three vertices u, v, w if uv and vw are arcs then uw is an arc. Every partial order $<$ on V can be represented as an oriented, transitive and acyclic oriented graph defined as follows: xy is an arc if and only if $x < y$. Note that a transitive tournament is the representation of a total order on V .

1.3 Parameterized complexity

In the following we introduce the parameterized complexity aspects developed throughout the manuscript. For more results concerning parameterized complexity, the reader is referred to the classical books [75, 93, 157].

A *decision problem* is a problem which returns either TRUE or FALSE. An instance of a decision problem is *positive* if its output is TRUE, otherwise, it is *negative*. A *parameterized problem* is a problem where the input is a pair (I, k) where I is the instance of the problem and k is an integer called the *parameter*. Usually, the parameter is either a part of the input (such as the size of the solution) or an invariant of the input graph (such as the treewidth). By abuse of notation, we will call parameters both the invariant which is the parameter and the size of this invariant.

Fixed Parameter Algorithms. A decision problem is *FPT* (or *Fixed Parameter Tractable*) according to a parameter k if there exists a constant c (which does not depend on k) and a computable function f such that for every instance of size n of parameter k , one can decide in time $f(k) \cdot n^c$ if the instance is positive or not. In other words, there exists an algorithm running in $f(k) \cdot n^c$ which solves the decision problem. Note that every polynomial time solvable problem admits FPT algorithms and the function f is indeed a polynomial function.

The goal of the parameterized complexity is twofold. The first one is theoretical: it permits to refine the class of NP-hard problems into several subclasses (which are hopefully disjoint). Indeed, several parameterized problems (probably) do not admit FPT algorithms. Consider the COLORING problem which given an integer k and a graph G , decides if there exists a proper k -coloring of G . The COLORING problem parameterized by k is not FPT. Indeed, deciding if a graph admits a proper 3-coloring is NP-complete (even for planar 4-regular graphs [67]). So there is no algorithm in $f(k)n^c$ algorithm to decide the COLORING problem, otherwise the complexity of 3-COLORING would be $f(3) \cdot n^c$, *i.e.* polynomial. In addition, several problems, such as k -CLIQUE, or k -DOMINATING SET, parameterized by the size of the solution (probably) do not admit FPT algorithms, even if they are polynomial time solvable when k is a fixed constant.

The second interest is more practical: it permits to tackle NP-complete problems and to obtain tractable algorithms when the parameter is small. So FPT algorithms provide efficient and exact algorithms for deciding NP-complete problems. In addition, parameterized complexity also permits to determine “why” an NP-complete problem is hard. Indeed, an FPT algorithm isolates a parameter from the remaining part of the instance in such a way that the unique exponential term of the complexity of the algorithm is due to this parameter. In some sense, the parameter is the core of the complexity of the problem.

Classical techniques. There are several classical techniques for proving that a problem is FPT. Let us describe some of them.

decision
problem
positive
negative
parameter

FPT

Branching algorithm. At each step of the algorithm, several choices are possible (and the instance is positive if and only if at least one of these choices leads to a positive instance). A branching algorithm “branches” over all these choices, which means that a branching algorithm runs distinct instances for every possible choice. Then we solve all the new, hopefully smaller, instances and return TRUE if and only if at least one of the new instances returns TRUE.

A branching algorithm can be represented as a search tree where each branching is a node of the tree and its sons are the new instances. The leaves are the instances which can be solved without new branchings. Note that the depth of the tree is the maximum number of consecutive branchings and the width of the tree is the maximum number of simultaneous branchings. If both width and depth of the branching tree are functions of k , then the whole tree has size $f(k)$ for some function f . So if the amount of work on each node is FPT, then the resulting algorithm is FPT. Let us illustrate the method on VERTEX COVER.

VERTEX COVER:

Input: A graph G , an integer k .

Output: TRUE if there exist k vertices which intersect all the edges of the graph, otherwise FALSE.

In other words, a vertex cover is a subset of vertices X such that $G[V \setminus X]$ is a stable set.

Theorem 1.8. VERTEX COVER is FPT parameterized by the size of the solution.

Algorithm 1: AlgoVC(G, k) for VERTEX COVER

Input : A graph $G = (V, E)$, an integer k .

Parameter: k .

Output : TRUE if there is a vertex cover of size at most k , FALSE otherwise.

```

1 if  $E = \emptyset$  then
2   | Return TRUE
3 if  $k = 0$  and  $E \neq \emptyset$  then
4   | Return FALSE
5 Let  $uv \in E$ ;
6 Boolean1:=AlgoVC( $G[V \setminus u], k - 1$ );
7 Boolean2:=AlgoVC( $G[V \setminus v], k - 1$ );
8 Return OR(Boolean1, Boolean2);

```

Sketch of the proof. Algorithm 1 describes a branching algorithm for solving VERTEX COVER. As long as there remains one edge uv in the graph, we branch in order to determine which vertex of u or v is in the vertex cover, which means that we “try” both choices (lines 6 and 7). At the end, the answer is positive if and only if one of the two branching instances returns TRUE (line 8). Note that if there is a solution of size at most k , then at least one of these two branches leads to a positive instance. Indeed every vertex cover contains either u or v since it covers the edge uv .

In order to determine if there is a vertex cover of size k , we just have to run Algorithm 1 with input (G, k) to decide if G has a vertex cover of size at most k . One can easily prove that Algorithm 1 has complexity $\mathcal{O}^*(2^k)$, where \mathcal{O}^* means that the polynomial term is not expressed. Indeed, at each

step, there are two possible choices, so the width of the branching tree is 2. In addition, the height of the tree is at most k since at each step the size of the parameter decreases by one and there is no branching when the parameter equals zero. \square

There are several other exponential time algorithms for VERTEX COVER, the best ones have complexity $\mathcal{O}^*(2^{1.27k})$ (see [46, 158]). The branching technique will be used in Chapter 6.

Iterative compression is another classical technique to design FPT algorithms. This technique has been introduced by Reed et al. [173] for designing an FPT algorithm for ODD CYCLE TRANSVERSAL. The general idea consists in using a solution of a smaller instance to find a solution of a larger instance. If the problem of deriving a solution from a solution of a smaller instance can be decided in FPT time, then the problem of finding a solution without any information can also be decided in FPT time. We will see a formal proof of this result for MULTICUT in Chapter 6.

Important separators and shadow removal. These techniques have been developed in the last few years and already have many applications. A large part of Chapter 2.3 is devoted to defining and illustrating these notions. Important separators are the core of the MULTICUT proof [33] presented in Chapter 6.

Other techniques. There are several other classical techniques for designing FPT algorithms. Color coding, introduced in [11], is a technique which consists in randomly coloring the graph. We then try to find a structure which intersects all colors, which is often simpler than to find an uncolored structure. Reduction to bounded treewidth graphs is also a classical method: it first consists in proving that a so-called irrelevant vertex can be found and deleted from the graph (in FPT time) if the treewidth is large enough. Then, one has to prove that the problem is FPT for bounded treewidth graphs, which is often simpler than for general graphs. A last technique for designing FPT algorithms consists in finding kernels. We will describe this technique in a further paragraph.

Parameters. In this paragraph we use the terms of treewidth and Monadic Second Order Logic, the reader is referred to [70, 93] for formal definitions. In the last few years, an important “ecology” of parameters has been developed. Let us describe the most standard ones in the next few lines. For more information, the reader is referred to [90].

The most classical parameter is the size of the solution. For instance, Theorem 1.8 ensures that VERTEX COVER parameterized by k (which is the size of the solution) admits an FPT algorithm. Then we refine the parameter k . By refining we mean that we can consider a weaker parameter (*i.e.* a parameter of smaller size) and determine if the problem is still FPT parameterized by this weaker parameter. Let α and β be two invariants such that for every graph G , we have $\alpha(G) \leq \beta(G)$. If the problem Π parameterized by α is FPT, then the problem Π parameterized by β is FPT. Indeed, since Π is FPT parameterized by α , the decision problem Π can be decided in $f(\alpha) \cdot n^c$. So it can also be decided in $f(\beta) \cdot n^c$ since $\alpha \leq \beta$. We say that the parameter α refines the parameter β .

The size of a vertex cover can be refined by the size of a feedback vertex set. Indeed the size of a vertex cover is at least the size of a feedback vertex set since a subset of vertices whose deletion eliminates all the edges of the graph in particular eliminates all the cycles of the graph. Since VERTEX COVER is FPT parameterized by the minimum size of a feedback vertex set by [121], VERTEX COVER is FPT parameterized by the size of a solution, *i.e.* [121] gives Theorem 1.8.

Note that a minimum vertex cover is the minimum number of vertices which have to be deleted in

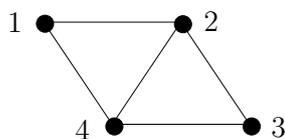


Figure 1.8: A graph which admits a vertex cover of size 2.

order to obtain a stable set, *i.e.* a graph of treewidth 0. So the minimum size of a vertex cover is the “distance to (graphs of) treewidth 0”. A minimum feedback vertex set is the minimum number of vertices which have to be deleted in order to obtain a forest. Since forests are graphs of treewidth at most 1, the size of a minimum feedback vertex set is the distance to treewidth 1. One can naturally generalize these two invariants by introducing the “distance to treewidth ℓ ” which is the minimum number of vertices whose deletion gives a graph of treewidth at most ℓ . It provides a hierarchy of parameters where every parameter refines the previous one. In other words, if a problem Π is FPT for the parameter distance to treewidth ℓ then Π is FPT for the parameter distance to treewidth ℓ' for every $\ell' \leq \ell$.

There is another classical parameter which is the treewidth of the input graph. A famous meta-theorem of Courcelle [63] ensures that every problem that can be expressed in Monadic Second Order Logic admits an FPT algorithm parameterized by the treewidth of the input graph. In addition the algorithm provided by Courcelle is linear when k is a constant (understand here that the degree c of the polynomial function in the FPT algorithm is 1). Nevertheless the exponential function in the treewidth is not tractable. Since VERTEX COVER can be expressed in Monadic Second Order Logic, VERTEX COVER is FPT parameterized by the treewidth of the input graph.

W-hierarchy. In order to prove that a problem (probably) does not admit a polynomial algorithm, the standard method consists in proving that the problem is NP-hard. Indeed a classical conjecture in complexity ensures that $P \neq NP$ and then NP-hard problems probably do not admit polynomial time algorithms. This conjecture has an “equivalent” for FPT algorithms. Let us first recall some definitions of logic. A *clause* is a disjunction of *literals* (which are positive or negative variables). The size of a clause is its number of literals. A *SAT formula* is a conjunction of clauses. A *2-SAT formula* is a conjunction of clauses of size 2. The sign \vee denotes the disjunction. We pay attention to the following problem:

WEIGHTED 2-SAT:

Input: 2-SAT formula ϕ , an integer k .

Parameter: k

Output: TRUE if there exists an assignment of the variables where at most k variables are assigned to true which satisfies the formula ϕ , otherwise FALSE.

Observation 1.9. WEIGHTED 2-SAT is NP-complete.

Sketch of the proof. VERTEX COVER is an NP-complete problem. We reduce VERTEX COVER to WEIGHTED 2-SAT. Let G be a graph. For every vertex u of the graph, create a variable x_u . For every edge uv , create the clause $x_u \vee x_v$. One can easily prove that the resulting 2-SAT formula has

a positive assignment with at most k positive variables if and only if G has a vertex cover of size at most k . \square

For the VERTEX COVER instance of Figure 1.8, the reduction algorithm creates a WEIGHTED 2-SAT instance with five clauses, $x_1 \vee x_2$, $x_1 \vee x_4$, $x_2 \vee x_4$, $x_2 \vee x_3$ and $x_3 \vee x_4$. Since all the clauses are satisfied when both x_2 and x_4 are assigned to true, 2 and 4 form a vertex cover of Figure 1.8. WEIGHTED 2-SAT is conjectured not to admit FPT algorithms. We say that the problem WEIGHTED 2-SAT is in $W[1]$ (introduced in [74]). A parameterized problem Π is $W[1]$ -hard if it is as complicated as WEIGHTED 2-SAT parameterized by k . By “as complicated as” we mean that there exists a reduction algorithm from every instance of WEIGHTED 2-SAT to a parameterized instance of Π such that:

- The complexity of the algorithm is FPT parameterized by k .
- The size of the parameter in the resulting instance of Π is a function of k .

On the opposite, a problem is in $W[1]$ if it can be reduced via an FPT parameterized reduction to WEIGHTED 2-SAT. A problem is $W[1]$ -complete if it is both in $W[1]$ and $W[1]$ -hard. The most classical $W[1]$ -complete problem is probably k -CLIQUE where k denotes the size of the desired clique. Lots of hardness reductions are reductions from k -CLIQUE in order to prove that a problem is $W[1]$ -hard (see [36] for instance).

The $W[1]$ -class can be extended into an infinite hierarchy of complexity classes called the W -hierarchy, which is the pendant of the polynomial hierarchy for the complexity NP-class. They satisfy

$$FPT \subseteq W[1] \subseteq W[2] \cdots \subseteq XP.$$

where XP denotes the set of parameterized problems which can be solved in $\mathcal{O}(n^{f(k)})$. In other words, XP is the set of problems which are polynomial when the parameter is not a part of the input but a constant.

Conjecture 2.

$$FPT \subsetneq W[1] \subsetneq W[2] \dots \subsetneq XP.$$

For more details on the $W[1]$ -hierarchy the reader is referred to [75].

Kernels. In parameterized complexity, there is another crucial notion which is the notion of kernels. In the following we assume that $k \leq n$ where n is the size of the instance and k the parameter. The parameterized problem Π admits a *kernel* (of size f) if there exists a polynomial time algorithm (in both the size of the instance and of the parameter) such that for every instance (I, k) , the algorithm returns an instance (I', k') of the problem Π such that:

- The new instance (I', k') satisfies $k' \leq k$.
- There exists a function f such that for every I , the output instance I' has size at most $f(k')$.
- The instance (I, k) of the problem Π is positive if and only if the instance (I', k') of the problem Π is positive.

Such an algorithm is called a *kernel algorithm*. Informally a kernel algorithm is an algorithm which reduces an instance of Π into a smaller instance (which only depends on the size of the parameter) which is positive if and only if the original instance is positive. The interest of a kernel algorithm is twofold. The first one is practical since kernel algorithms provide interesting pre-processing algorithms. Indeed they transform arbitrarily large instances into instances whose size only depends

$W[1]$ -
complete

kernel

on k in polynomial time. The second interest is theoretical: if a problem admits a kernel, then, since this kernel can be obtained in polynomial time, it means that all the hardness of the problem is reduced to this kernel. It raises a natural question: up to which size can we reduce the size of the problem?

Existence of kernels and existence of FPT algorithms are actually equivalent.

Lemma 1.10 (Folklore). *A problem Π admits an FPT algorithm if and only if it admits a kernel.*

Proof. Assume that Π admits a kernel algorithm \mathcal{A} whose complexity is $P(n)$ where P is a polynomial function, and n denotes the size of the instance. Let (I, k) be an instance of Π . The algorithm \mathcal{A} on the instance (I, k) returns an instance (I', k') equivalent to (I, k) of size $f(k')$ where $k' \leq k$. Then decide with any algorithm if the instance (I', k') is positive or not. Note that the complexity of this second algorithm is a function say g of k' since the size of the instance I' is a function of k' . Therefore the whole complexity is $P(n, k) + g(k') = P(n) \cdot f(k)$ since $k' \leq k$.

Let us prove the converse. Since Π is FPT, an algorithm \mathcal{A} decides Π in time $f(k) \cdot P(n)$. Let us prove that Π admits a kernel of size $f(k)$. Let (I, k) be an instance of the parameterized problem Π . First note that if $|I| \leq f(k)$ then no kernel algorithm is needed since the size of the instance is at most the size of the desired kernel. Otherwise, $|I| \geq f(k)$, so the algorithm \mathcal{A} , running in time $f(k) \cdot P(n) \leq nP(n)$, is a polynomial time algorithm for (I, k) . So we can decide the instance (I, k) in polynomial time and return a trivial instance which is either positive if $\Pi(I, k)$ is positive or negative otherwise. Such an algorithm is a kernel algorithm since its running time is polynomial. \square

A much more constrained condition is the existence of polynomial kernels. A problem admits a *polynomial kernel* if the function f of the kernel definition is a polynomial function. Finding small kernel is interesting for several reasons. First, it can improve FPT algorithms. Indeed if the kernel is small enough and if the exponential algorithm is good enough, the resulting FPT algorithm can be efficient. For instance, MAXIMUM INTERNAL SPANNING TREE problem has a polynomial kernel of size $3k$ [94] and an exact exponential algorithm of complexity $\mathcal{O}^*(2^n)$ [156], which provides a $\mathcal{O}^*(2^{8k})$ exponential time algorithm [94] improving the older FPT algorithms.

polynomial
kernel

Note nevertheless that all the FPT problems do not admit polynomial kernels unless $NP \subseteq coNP/poly$. A machinery was developed in order to determine kernel lower bounds, such as OR-compositions, and more recently CROSS-compositions [23, 24]. So, problems with polynomial kernels refine the class of FPT problems since there are problems which admit polynomial kernels and several problems which do not admit polynomial kernel.

We have now defined the main combinatorial and algorithmic concepts needed to tackle more precisely the study of hitting sets and packings. The following Chapter is devoted to the general study of hitting sets and packings.

Hitting sets

In this chapter we introduce the notions of hitting sets and packings. The results mentioned all along this chapter are general results on hypergraph theory (and none of them were proved during my PhD). The chapter is organized in such a way the introduced notions are more and more involved.

2.1 Hitting sets and packings

This Chapter is an introduction on hitting sets and packings in hypergraphs. In Section 2.1, we introduce these two notions, provide several examples and study the gap between the two associated parameters. In Section 2.2, we will first express these two notions in terms of linear programming. We will see that these two notions are dual, not in the hypergraph meaning introduced in Chapter 1 but in the linear programming meaning. In order to avoid confusion, the duality between hitting set and packing number will be referred as LP-duality. We will finally show that the so-called “integrality gap” between these notions and their fractional relaxations can be arbitrarily large. In Section 2.3, we will consider algorithmic aspects of hitting sets through the eyes of graph separation problems. In particular we study two keys methods for designing FPT algorithms for graph separation problems which are the important separators technique (Section 2.3.2, a tool introduced by Marx) and the shadow removal technique (Section 2.3.4, a tool introduced by Marx and Razgon). These two techniques are technically very involved. Since we study them in details in Sections 2.3.2 and 2.3.4, these two sections are much more involved than the other part of the chapter.

2.1.1 Definitions and first properties

Let $H = (V, E)$ be a hypergraph with no empty hyperedge. A *hitting set* of H , also called a *transversal*, is a subset X of vertices such that, for every hyperedge $e \in E$, at least one vertex of X is in e . In other words, a hitting set is a subset of vertices intersecting all the hyperedges. Note that the whole set of vertices V is a hitting set. In Figure 2.1, the set of gray vertices is a hitting set of size 3. The *transversality* of H , denoted by $\tau(H)$ is the minimum size of a hitting set of H .

hitting set
transversal

transversality
 τ

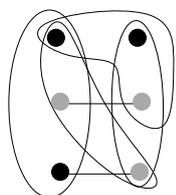


Figure 2.1: The set of gray vertices is a hitting set of size 3.

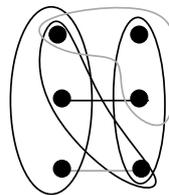


Figure 2.2: The set of gray hyperedges is a packing of size 2.

Determining if a hypergraph H has a hitting set of size at most k is an NP-hard problem, even for 2-uniform hypergraphs (one of the 21 Karp's NP-complete problems called VERTEX COVER).

A subset of hyperedges $P \subseteq E$ is a *packing* of H if every vertex of H is in at most one hyperedge of P . In other words, all the hyperedges of P are vertex disjoint. In Figure 2.2, the set of gray hyperedges is a packing of size 2. The *packing number* of H , denoted by $\nu(H)$, is the maximum size of a packing of H . Note that a hypergraph H satisfies $\nu(H) = 1$ if all its hyperedges pairwise intersect. Determining if a hypergraph has a packing of size k is an NP-complete problem, even for 3-uniform hypergraphs (the problem is the 3-dimensional matching problem). Nevertheless for 2-uniform hypergraphs the problem is polynomial time solvable (indeed a packing of a 2-uniform hypergraph is exactly a matching of the corresponding graph and a maximum matching can be found in polynomial time). When no confusion is possible, transversality and packing number will be denoted by τ and ν instead of $\tau(H)$ and $\nu(H)$.

Observation 2.1. *Every hypergraph with no empty hyperedge H satisfies*

$$\nu(H) \leq \tau(H).$$

Proof. Let P be a packing and X be a hitting set of H . Every hyperedge e of P satisfies $e \cap X \neq \emptyset$. And no vertex of X intersects more than one edge of the packing since $e \cap e' = \emptyset$ for every $e, e' \in P$. So $|X| \geq |P|$. The desired inequality is obtained by considering a maximum packing and a minimum hitting set. \square

Observation 2.1 raises a natural question: is τ also bounded by a function of ν ? Unfortunately the answer is NO. Consider the complete uniform hypergraph $\mathcal{U}_{n+1, 2n}$. Its packing number equals one. Indeed every hyperedge contains more than half the vertices, so every pair of hyperedges intersects. On the contrary, the transversality of $\mathcal{U}_{n+1, 2n}$ is at least n . Otherwise the complement of a hitting set would have size at least $n + 1$, and then would contain a hyperedge (since every subset of size $n + 1$ is a hyperedge), a contradiction.

Nevertheless, in several cases, τ can be bounded by a function of ν . In the following we present several classes of hypergraphs satisfying this property.

Erdős-Pósa property and feedback vertex set. Let $G = (V, E)$ be a graph. Recall that a feedback vertex set of G is a subset of vertices X such that $G[V \setminus X]$ is acyclic. Note that it also is a subset of vertices intersecting all the cycles of G . Indeed, every cycle of G contains a vertex of a feedback vertex set X , since otherwise $G[V \setminus X]$ would contain this cycle (and then the resulting graph would

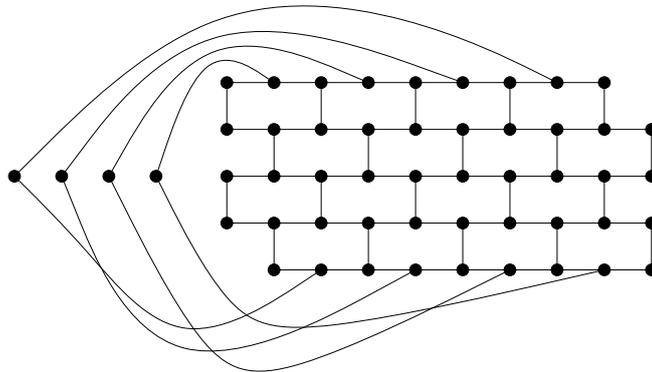


Figure 2.3: An Escher's wall of height 4. To increase the height of the Escher's wall by one, add a line and a brick per line (and a new vertex at the left). One can prove that every odd cycle passes through one of the left vertices so we have $\nu = 1$. And one can find $h/2$ odd cycles such that every vertex is in at most two odd cycles (where h denotes the height of the wall). So $\tau \geq 1/2 \cdot h/2$.

not be acyclic). Determining the size of a minimum feedback vertex set problem is NP-hard even for graphs of maximum degree 4 [175]. In their seminal paper [82], Erdős and Pósa proved the following.

Theorem 2.2 (Erdős, Pósa [82]). *Let G be a graph. Denote by τ the size of a minimum feedback vertex set and by ν the maximum number of vertex disjoint cycles. We have*

$$\tau = \mathcal{O}(\nu \cdot \log \nu).$$

In addition Erdős and Pósa proved that the upper bound can be achieved on random graphs, using a probabilistic argument [82].

The cycle hypergraph H of the graph G has vertex set $V(G)$ and $C \subseteq V$ is a hyperedge if $G[C]$ is a (non-necessarily induced) cycle. A packing of H is a subset of vertex disjoint hyperedges, so, in the graph G , it corresponds to a collection of vertex disjoint cycles. A hitting set of H is a subset of vertices intersecting every hyperedge, *i.e.* every cycle of G . So a hitting set of H is a feedback vertex set of G . Hence Theorem 2.2 can be rephrased as follows: cycle hypergraphs satisfy $\tau = \mathcal{O}(\nu \cdot \log \nu)$. A class of hypergraphs with $\tau \leq f(\nu)$ satisfies the Erdős-Pósa property. More formally, let \mathcal{H} be a family of hypergraphs. The family \mathcal{H} satisfies the *Erdős-Pósa property* (for the *gap function* f) if for every hypergraph H in \mathcal{H} , we have $\tau(H) \leq f(\nu(H))$. Note that a class of hypergraphs \mathcal{H} satisfying $\nu = 1$ satisfies the Erdős-Pósa property if and only if there exists a constant c such that every $H \in \mathcal{H}$ satisfies $\tau(H) \leq c$.

Erdős-Pósa
property
gap function

Theorem 2.2 was generalized by Kakimura, Kawarabayashi and Marx who proved that the set of cycles intersecting a fixed set S also has the Erdős-Pósa property [123]. Pontecorvi and Wollan closed the gap in [166] between upper and lower bound by giving an $\mathcal{O}(\nu \cdot \log \nu)$ upper bound. The result of Erdős and Pósa was also generalized to several other types of cycles. Indeed given a particular set of cycles, one can ask if this set of cycles satisfies the Erdős-Pósa property. Even cycles satisfy the Erdős-Pósa property [183] as planar minors [50, 176]. On the contrary, Lovász and Schrijver noticed that odd cycles do not satisfy the Erdős-Pósa property. Escher's walls give a family of graphs

with packing number one and with arbitrarily large transversality (see Figure 2.3). For a complete proof of this result see [169] for instance. The existence of an equivalent version of Theorem 2.2 for directed graphs, known as Gallai-Younger conjecture, was solved by Reed et al. in [172].

k -uniform hypergraphs. Let us first illustrate the notions of packings and hitting sets on uniform hypergraphs.

Observation 2.3. *Let $k > 0$. Every k -uniform hypergraph H satisfies*

$$\tau(H) \leq k \cdot \nu(H).$$

Proof. Consider a maximal (in terms of inclusion) packing P of H . Let $X = \bigcup_{e \in P} \bigcup_{x \in e} x$. In other words X is the set of vertices contained in the hyperedges of P . Since every hyperedge has size k , the set X has size at most $k \cdot \nu$. Let us show that the set X is a hitting set of H . Assume by contradiction that a hyperedge e satisfies $e \cap X = \emptyset$. Then no vertex of e is in a hyperedge of P , and e can be added in the packing P , which contradicts the maximality of P . \square

The inequality $\tau \leq k \cdot \nu$ provided by Observation 2.3 is tight (up to an additive constant). Indeed consider the complete uniform hypergraph $\mathcal{U}_{k,n}$. A maximal packing P has size at most $\lfloor n/k \rfloor$. Indeed $\sum_{e \in P} |e| \leq n$ since every vertex appears in at most one hyperedge of the packing P . Since every hyperedge e satisfies $|e| = k$, we have $|P| \leq n/k$. A hitting set has size at least $n - k + 1$. Otherwise, at least k vertices would not be in the hitting set, and then a hyperedge would not be intersected by the hitting set (since every subset of size k is a hyperedge), a contradiction. Finally we have $\tau + k - 1 \geq n \geq k \lfloor n/k \rfloor \geq k\nu$.

The proof of Observation 2.3 leads to a k -approximation algorithm for k -uniform hypergraphs. Greedily add hyperedges in a packing P as long as possible. At the end of the algorithm, denote by $X = \bigcup_{e \in P} \bigcup_{x \in e} x$. Since P is a packing, X has size at most $k \cdot \nu(H)$. And since P is maximal by inclusion, X is a hitting set. Observation 2.1 ensures that $\nu \leq \tau$, so X is a hitting set of size at most $k \cdot \tau$. Therefore the greedy algorithm is a k -approximation algorithm.

Consider the particular case of graphs (which are 2-uniform hypergraphs). Let G be a graph. A *vertex cover* is a hitting set of G . Determining the minimum size of a vertex cover is NP-hard. A *matching* is a packing of G . On the contrary, determining the size of a maximum size of a matching in a graph can be done in polynomial time (see [79] for instance). The following result due to Konig links the size of a minimum vertex cover and the size of a maximum matching in bipartite graphs.

Theorem 2.4 (Konig [134]). *The size of a maximum matching equals the size of a minimum vertex cover in bipartite graphs.*

The equality of Theorem 2.4 is false on general graphs. Indeed, the clique K_n has matchings of size at most $n/2$ and vertex covers of size at least $n - 1$. Ryser's conjectured that Theorem 2.4 can be extended to k -partite hypergraphs. A hypergraph is *k -partite* if there exists a partition of the vertex set into V_1, \dots, V_k such that every hyperedge contains exactly one vertex in each set.

Conjecture 3 (Ryser). *Every k -partite hypergraph satisfies $\tau \leq (k - 1)\nu$.*

vertex cover
matching

k -partite

Conjecture 3 generalizes Theorem 2.4. Indeed, in the particular case $k = 2$, Observation 2.1 ensures that $\nu \leq \tau$, so if $\tau \leq (k - 1)\nu$ then $\tau = \nu$. Aharoni proved Conjecture 3 for $k = 3$ [2]. In the specific case $\nu = 1$, Tuza [187] proved Conjecture 3 for $k = 4$ and $k = 5$ (the problem is still open for $k \geq 6$). Haxell and Scott generalized Tuza's result by proving that $\tau \leq (k - \epsilon)\nu$ [118] for some positive constant ϵ for $k = 4, 5$. In the general case Ryser's conjecture is still open for $k \geq 4$.

Menger's theorem. Another well-known theorem of graph theory can be rephrased as an Erdős-Pósa property: Menger's theorem. Let G be a graph and s, t be two vertices of G . An st -path is a path from s to t in G . An (edge) st -separator is a subset S of edges whose deletion puts s and t in distinct connected components. In other words, S is a set of edges intersecting all the st -paths. Indeed if an st -path is not intersected by S then both s and t are in the same connected component.

st -path
 st -separator

The st -transversal hypergraph of the graph G is the hypergraph on the edges of the graph G where hyperedges are the edges of st -paths. A hitting set of this hypergraph is a subset of edges which intersects every hyperedge, so every st -path. Hence a hitting set is a subset of edges whose deletion disconnects s from t . Conversely, one can easily verify that every st -separator is a hitting set of the st -transversal hypergraph. So the transversality of the st -transversal hypergraph is the minimum size of a st -separator. And the packing number denotes the maximum number of disjoint hyperedges, so it is the maximum number of interior disjoint paths between s and t . All the previous definitions also holds if we replace s and t by two sets A and B . An AB -path is a path from a vertex of A to a vertex of B . And an AB -separator is a subset of vertices which intersects every path between a vertex of A and a vertex of B . A famous theorem due to Menger (whose proof can be found in [70] for instance) states the following:

Theorem 2.5 (Menger). *Let G be a graph and A, B be two subsets of vertices. The minimum size of an AB -separator equals the maximum number of edge disjoint AB -paths, i.e. the AB -transversal hypergraph satisfies $\tau = \nu$.*

In addition, Menger's theorem proofs provide a polynomial time algorithm to find minimum separators and maximum collections of edge disjoint paths in polynomial time. Generalizations of Menger's theorem have been extensively studied. If, instead of looking for separators between two vertices, we are looking for a minimum separator between 3 vertices, the problem becomes NP-complete. Such problem, known as MULTIWAY CUT, is a well-studied graph separation problem. We will discuss this problem and its generalizations a little bit further in Section 2.3 and Chapter 6.

Note finally that all along this paragraph we deal with edge separators. Nevertheless Menger's Theorem also holds from a vertex point of view. In other words, the maximum set of vertex disjoint paths between s and t equals the minimum number of vertices which have to be deleted in order to put s and t in distinct connected components.

2.1.2 Helly property

Informally, a family of sets satisfies the *Helly property* if every subfamily of pairwise intersecting sets has an element in all of them. Helly property often holds for families of geometrical objects (where sets are viewed as sets of points of the space). For instance, intervals of the line satisfy the Helly property. This example can be extended to d -dimensional spaces with axis-parallel d -

Helly
property

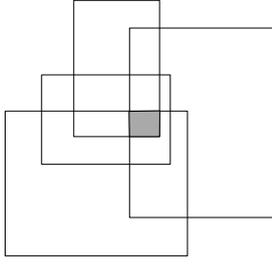


Figure 2.4: Rectangles of the plane satisfy the Helly property: the gray part is in the intersection of all the rectangles.

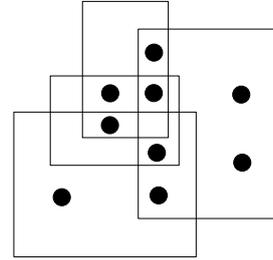


Figure 2.5: On the contrary, hypergraphs of rectangles do not satisfy the Helly property.

dimensional rectangles. Figure 2.4 represents pairwise intersecting rectangles: the gray part is in the intersection of the collection of the set of rectangles.

A hypergraph has the *Helly property* if for every subset E' of pairwise intersecting hyperedges, there exists a vertex v of V such that $v \in \bigcap_{e \in E'} e$. Hypergraphs can represent objects in geometrical spaces. Consider for instance a set V of points of the plane and \mathcal{R} a collection of rectangles of the plane. It naturally induces a hypergraph where the vertex set is V and for every $R \in \mathcal{R}$, there is a hyperedge e_R such that $v \in e_R$ if and only if v is in the rectangle R (see Figure 2.5). The hypergraph can be represented in the plane with the same set of rectangles. Geometrical hypergraphs are built from geometrical problems, but even if the Helly property is satisfied for the geometrical problem, it is not necessarily satisfied for the hypergraph problem. Figure 2.5 illustrates this phenomenon. Indeed, every pair of hyperedges (which correspond to rectangles in the plane) pairwise intersects in H , but since there is no vertex in the area of intersection of the whole set of rectangles. Hence the set of hyperedges does not intersect in the hypergraph: the hypergraph does not satisfy the Helly property.

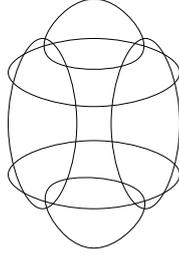
If a Helly hypergraph satisfies $\nu = 1$, then $\tau = 1$. Indeed, since the packing number equals one, all the hyperedges pairwise intersect. And then the Helly property ensures all the hyperedges intersect on a same vertex. Nevertheless, and quite surprisingly at first glance, Helly hypergraphs do not have the Erdős-Pósa property as underlined in [114].

Lemma 2.6. *There exist Helly hypergraphs H_k such that $\nu(H_k) = 2$ and $\tau(H_k) = k$.*

Proof. Let G_k be a k -chromatic graph of girth at least 4 (such graphs exist as mentioned in Chapter 1). Construct the stable set hypergraph H_k of G_k . Recall that H_k is constructed as follows: the vertices of H_k are maximal (by inclusion) stable sets of H . For every vertex x of G_k , construct a hyperedge containing all the stable sets S of $V(H_k)$ such that $x \in S$. In the following, we denote by e_x the hyperedge of H_k corresponding to x in G_k . Let us first prove the following claim.

Claim 2.7. *For every x, y in $V(G_k)$, we have $e_x \cap e_y = \emptyset$ if and only if xy is an edge.*

Proof. If xy is an edge, then no stable set contains both x and y , so $e_x \cap e_y = \emptyset$. Conversely, if xy is not an edge, then $\{x, y\}$ is a stable set. It can be completed into a stable set S maximal by inclusion and $S \in e_x \cap e_y$. \square

Figure 2.6: A hypergraph satisfying the $(3, 2)$ -property.

Let us prove that H_k is a Helly hypergraph. Let $e_{x_1}, \dots, e_{x_\ell}$ be pairwise intersecting hyperedges. By Claim 2.7, for every pair $i \neq j$, $x_i x_j$ is not an edge. So $\{x_1, \dots, x_\ell\}$ is a stable set which can be completed into a stable set S maximum by inclusion. Finally we have $S \in \cap_{i \leq \ell} e_{x_i}$.

Let us now evaluate ν and τ . If there is a packing of size 3, Claim 2.7 ensures that G_k contains a clique of size 3, which contradicts our girth assumption. Indeed a graph of girth at least 4 cannot contain a triangle (which is a cycle of length 3). So the packing number is at most 2. On the contrary, $\tau(H_k) \geq k$. A hitting set \mathcal{S} is a family of stable sets of G_k intersecting all the hyperedges of H_k . So for every vertex x , $e_x \cap \mathcal{S} \neq \emptyset$. In other words, every vertex of G_k appears in a stable set of \mathcal{S} . So $|\mathcal{S}|$ stable sets cover all the vertices of G_k . Observation 1.7 ensures that $|\mathcal{S}| \geq \chi(G_k) = k$. Finally every hitting set has size at least k , *i.e.* we have $\tau \geq k$. \square

Note that the proof of Lemma 2.6 ensures that the stable set hypergraph of any graph has the Helly property. Let us provide another simple example of a non-geometrical hypergraph with the Helly property.

Observation 2.8. *Let T be a tree. Any subset of subtrees of T satisfies the Helly property.*

Proof. Let us prove it by induction on the number of nodes of the tree. Let \mathcal{T} be a family of pairwise intersecting subtrees of T (where subtrees can be single vertices). Let f be a leaf of T . Either a subtree is reduced to the leaf f , and since subtrees of \mathcal{T} pairwise intersect, all the subtrees contain f , *i.e.* $\tau = 1$. Otherwise we apply induction on $T[V \setminus \{f\}]$ (the family still pairwise intersects since every subtree containing f also contains its unique neighbor since a tree is a connected graph). \square

(p, q) -property. A hypergraph has the (p, q) -property if for every set of p hyperedges, at least q of them intersect. Figure 2.6 represents a set of hyperedges satisfying the $(3, 2)$ -property since for every set of three hyperedges, at least two of them intersect. Note that a hypergraph H satisfies the $(p, 2)$ -property if and only if $\nu < p$. Indeed a hypergraph with the $(p, 2)$ -property satisfies that for every subset of p hyperedges, at least one vertex is in two hyperedges, *i.e.* there are no p vertex disjoint hyperedges. Hadwiger and Debrunner asked for the existence of a function $M(p, q, d)$ such that every set of compact convex sets in \mathbb{R}^d with the (p, q) -property satisfies $\tau \leq M(p, q, d)$ [115]. Such a statement generalizes Helly's theorem [119] which ensures that compact convex sets in \mathbb{R}^d with the $(d + 1, d + 1)$ -property satisfy $\tau = 1$. Alon and Kleitman proved the Hadwiger-Debrunner's conjecture [8]. In Chapter 3, we will see that under some combinatorial properties, the (p, q) -property implies the Erdős-Pósa property.

(p, q) -
property

2.1.3 Duality of hypergraphs

Let us study the behavior of hitting sets and packings through the eyes of duality of hypergraphs. Recall that in the dual hypergraph, vertices become hyperedges and hyperedges become vertices (for a formal definition, see Section 1.1).

covering

Duality of τ : covering. Given a hypergraph H , a *covering* E' of H is a subset of edges such that every vertex is contained in at least one hyperedge of E' . The *covering number*, denoted by $c(H)$ is the minimum size of a covering. Note that a covering exists if and only if for every vertex, there exists a hyperedge containing it. For graphs (understand here 2-uniform hypergraphs), the covering number is crucially linked with the size of a maximum matching. More precisely, the covering number of a graph equals n minus the size of a maximum matching. Indeed, each edge of a maximum matching covers two distinct (new) vertices. To cover the remaining vertices, we add edges which only cover one vertex not yet covered. Since the size of a maximum matching in a 2-hypergraph is exactly the maximum number of disjoint hyperedges, we have

$$c(G) = n - \nu(G).$$

A matching is *perfect* if every vertex of the graph is covered by the matching (so $\nu = c$). Note that only graphs with an even number of vertices can admit perfect matchings and their sizes are exactly $n/2$. The existence of perfect matchings in graphs has been extensively studied in the literature. For instance, bridgeless cubic graphs admit perfect matchings (Petersen's theorem, see [70] for a proof).

Observation 2.9. *For every hypergraph H with no empty hyperedge, the dual of a hitting set is a covering. In particular, we have*

$$\tau(H) = c(H^d).$$

Proof. Let H be a hypergraph and X be a hitting set of H . Let $((V, E), E')$ be the incidence bipartite graph of H . Let e be a hyperedge of H . Since X is a hitting set $e \cap X \neq \emptyset$. So there exists $x \in X$ such that $x \in e$, i.e. xe is an edge of the incidence bipartite graph. Hence the neighborhoods of the vertices of X cover the vertices of E in the incidence bipartite graph. The same holds in the dual incidence bipartite graph since the edges are not modified. Finally, in the dual hypergraph X is a subset of hyperedges containing all the vertices.

The reverse inequality is given by Observation 1.2 which ensures that $(H^d)^d = H$. \square

stable set

Duality of ν : stable set. Given a hypergraph, a *stable set* is a subset of vertices such that no hyperedge contains at least two of them. The size of a maximum stable set is denoted by α . Note that a stable set of a 2-uniform hypergraph is a subset of vertices X such that no hyperedge contains two vertices of X , i.e. no hyperedge is contained in X since hyperedges have size 2. So the notion of stable set in hypergraph extends the notion of stable sets in graphs.

Observation 2.10. *For every hypergraph H , the dual of a packing is a stable set. In particular, we have*

$$\nu(H) = \alpha(H^d).$$

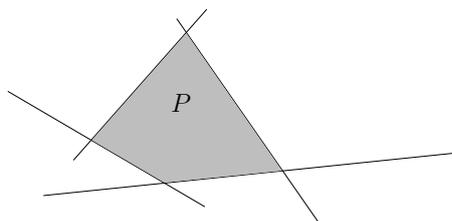


Figure 2.7: A 2-dimensional polytope defined by 4 constraints.

Proof. Let E' be a packing of a hypergraph H . In the dual hypergraph, E' becomes a subset of vertices. Since E' is a packing, no vertex of H is contained in two hyperedges of E' . Therefore, in the dual hypergraph no hyperedge contains at least two vertices of the corresponding subset of vertices E' . □

2.2 Linear programming and integrality gap

In this Section, we introduce general notions of linear programming and apply them on hitting sets. For more information and complete proofs, the reader is referred to [180].

2.2.1 Generalities

Half-spaces and polyhedra. Let us first recall classical definitions of linear algebra. The set \mathbb{R}^d denotes the real d -dimensional space. A *vector* is a ordered set of d reals. It can be seen as the coordinates of a point in \mathbb{R}^d . Let X be a set of ℓ vectors of \mathbb{R}^d . The *cone generated by X* is the set of all the positive combinations of points of X . In other words, the cone generated by X is $\{\sum_{x \in X} \alpha_x x \mid \alpha_x \geq 0\}$.

The *transposition of a matrix A* is denoted by ${}^t A$ and the coefficient (i, j) of the matrix ${}^t A$ is the coefficient (j, i) of A . Note that the transposition of a $n \times d$ matrix is a $d \times n$ matrix. The (affine) *hyperplane* defined by a in \mathbb{R}^d and $\beta \in \mathbb{R}$ is the set of points of \mathbb{R}^d such that ${}^t a x = \beta$. The *half-space* defined by a vector a of \mathbb{R}^d and $\beta \in \mathbb{R}$ is the subset of points x of \mathbb{R}^d such that ${}^t a x \leq \beta$. From a geometrical point of view, an half-space is the set of points of \mathbb{R}^d which are below (or above) a hyperplane.

A *polyhedron* is an intersection of several half-spaces. A *polytope* is a bounded polyhedron. The gray part of Figure 2.7 is a polytope of a 2-dimensional space. Since every half-space can be expressed as a linear inequation ${}^t a x \leq b$, every intersection of half-spaces ${}^t a_i x \leq b_i$ can be transformed into a matrix inequation $Ax \leq b$. The i -th line of the matrix is ${}^t a_i$ and the i -th coordinate of b is b_i . So every polyhedron can be expressed as the set of points x such that $Ax \leq b$ where A is an $n * d$ matrix, where n is the number of half-spaces and d is the dimension of the space. Recall that $Ax \leq b$ is satisfied if the inequality is satisfied on each coordinate. In the following every half-space intequality will be called a *constraint*.

Linear programming. A *linear program*, abbreviated into LP, consists in maximizing or minimizing a linear function over a polyhedron. The function we want to maximize (or minimize) is called

vector

hyperplane
half-space

polyhedron
polytope

constraint

LP

the *objective function*. The polyhedron can be represented with a matrix A called the *constraint matrix* and a vector b called the *constraint vector*. A *solution* of a linear program is a vector which satisfies the constraints, in other words, a solution is a point in the polyhedron. Over all the solutions, we want to maximize (or minimize) an objective linear function denoted by ${}^t c x$ for some vector c . By abuse of notation, c is also called the objective function. For summary, a linear program can be seen as the following value:

$$\max_{x \in \mathbb{R}^d} ({}^t c x \mid Ax \leq b).$$

The *optimal value* is the real number corresponding to the linear program. A solution x of the linear program is *optimal* if the objective function on x has the optimal value.

From a geometrical point of view, the objective function is a direction in \mathbb{R}^d , and an optimal solution is a point of the polyhedron which is the furthest in this direction (see Figure 2.8 for an illustration).

There are two famous algorithms for computing optimal solutions of LP. The first one, due to Dantzig, is called the simplex algorithm [68]. It is a non-polynomial (in the worst case) algorithm but it is really efficient in practice. In the 80's, a polynomial time algorithm, based on the so-called *ellipsoid method*, was discovered (see [109] for complete information on the algorithm).

Theorem 2.11 (Ellipsoid method). *An optimal solution of a (real) LP can be found in polynomial time.*

Let $\max({}^t c x \mid Ax \leq b)$ be a linear program. The *LP-dual* of the linear program is $\min({}^t y b \mid y \geq 0, {}^t y A = {}^t c)$. Note that the LP-dual of a linear program is a linear program. The initial linear program is called the *primal linear program*. The LP-duality theorem ensures the following :

Theorem 2.12 (Duality of LP). *For every linear program over real numbers (with a finite optimal value), we have*

$$\max({}^t c x \mid Ax \leq b) = \min({}^t y b \mid y \geq 0, {}^t y A = {}^t c).$$

Note that a linear program can admit no solution. In this case we assume the following: $\min \emptyset = +\infty$ and $\max \emptyset = -\infty$. As we have already seen, each line of the constraint matrix “represents” a constraint. In addition each column “represents” a variable since for each constraint, the value of the i -th column still multiplies the i -th variable. In the dual linear program, since we multiply by the left hand side the matrix A instead of the right hand side, the variables “are” the lines of the matrix A and the constraints “are” the columns of the matrix A . So variables of the dual LP represent constraints of the primal LP. And constraint of the dual LP represent variables of the primal LP. Theorem 2.12 ensures that there is an equality between primal and dual linear program in real numbers. This equality which is correct for real numbers is not correct for integer linear programs.

Integer Linear Program. Up to this point, we have only considered linear program in real spaces. Nevertheless, in several cases we are looking for integer solutions, *i.e.* a point which is in the polyhedron, which has integer coordinates and which maximizes the objective function according to these two conditions. In other words, we consider the intersection of the polyhedron with the integer grid in the space \mathbb{R}^d and we try to maximize a linear function. To sum up, an *integer linear program* (ILP for short) has the same constraint matrix and objective function as a LP program ex-

objective
function
constraint
matrix

ellipsoid
method

LP-dual

primal

LP

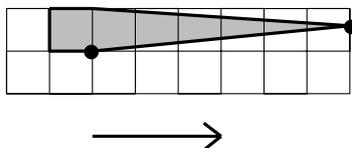


Figure 2.8: Gap between LP and ILP: the arrow represents the objective function and the gray part the polyhedron. The rightmost point is an optimal real solution and the leftmost is an optimal integer solution (integer values are represented with the grid).

cept that the variables must take integer values instead of real values. When we are given an ILP, the corresponding real number LP is called the *fractional relaxation* of the integer linear program.

The gap between integer and real optimal values of a LP is called the *integrality gap*. In the following we will say that integrality gap is bounded if there exists a function f such that $\tau \leq f(\tau^*)$. We will say that there is no integrality gap if $\tau = \tau^*$. The integrality gap can be arbitrarily large. Figure 2.8 represents a polytope with a large gap between the optimal integral solution and the optimal real solution. One can easily modify this example in order to have an arbitrarily large gap. In Section 2.2.2, we will study the integrality gap of the transversality and the packing LP. Even if computing the fractional relaxation of an ILP can be done in polynomial time by Theorem 2.11, computing the optimal value of an ILP is an hard problem in general (we will see in Section 2.2.2 that the transversality which is NP-hard to compute can be expressed as an ILP). Note that, in several cases, the integrality gap is bounded or, even better, there is no integrality gap.

fractional re-
laxation
integrality
gap

2.2.2 Transversal and Packing Linear Programs

In the following, we will consider the transversality and the packing number as linear programs. Let $H = (V, E)$ be a hypergraph.

Transversal (Integer) Linear Program:

Variables: A variable $x_v \in \mathbb{R}$ (or \mathbb{N}) for every $v \in V$.

Constraints: For every hyperedge $e \in E$, $\sum_{v \in e} x_v \geq 1$.
For every vertex v , $x_v \geq 0$.

Objective function: minimize $\sum_{v \in V} x_v$.

The constraint matrix of the Transversal LP of the hypergraph of Figure 2.9 is represented in Figure 2.10. Note that there are two types of constraints. The *hyperedge constraints* are the constraints on the hyperedges of H . The *positivity constraints* are the constraints on the vertices of H . There exists a bijection between the set of variables and the set of vertices of the hypergraph. By abuse of notation, and when no confusion is possible, we will say that the variables are the vertices of the hypergraph. The values of the variables can be seen as a weight function on the vertices of the hypergraph. A weight function $w : V \rightarrow \mathbb{R}$ satisfies the transversal linear program if, when we give the value $w(v)$ to the variable x_v then all the constraints are satisfied. We will denote by $w(V)$ the sum of the weights of all the vertices.

Observation 2.13. *Let H be a hypergraph with no empty hyperedges. The optimal value of the Transversal Integer Linear Program of H equals $\tau(H)$.*

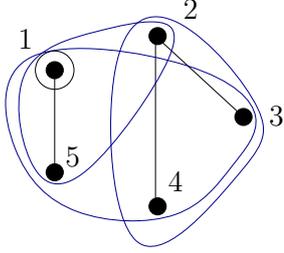


Figure 2.9: A hypergraph. For readability the hyperedges of size at least 3 are represented in blue.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 2.10: Constraint matrix A of the Transversal LP of Figure 2.9. The seven first lines are hyperedges constraints. The five others are positivity constraints.

Proof. Let w be an optimal weight function $V \rightarrow \mathbb{N}$ of the Transversal ILP. First note that each variable has either value zero or one. Indeed, assume by contradiction that some vertex v satisfies $w(v) \geq 2$. Consider the weight function w' where $w'(u) = w(u)$ for every $u \neq v$ and $w'(v) = w(v) - 1$. All the constraints are still satisfied. For the positivity constraints, it is immediate. For the hyperedge constraints, either $v \notin e$ and then we have $\sum_{u \in e} w'(u) = \sum_{u \in e} w(u) \geq 1$ since the function w is a solution of the transversal LP. Or $v \in e$, and then we have $\sum_{u \in e} w'(u) \geq w'(v) \geq 1$. So w' is also a solution of the transversal LP and $\sum_{u \in V} w'(u) < \sum_{u \in V} w(u)$, a contradiction with the optimality of w .

So w is a function $V \rightarrow \{0, 1\}$. Let X be the subset of vertices v such that $w(v) = 1$. Since $\sum_{u \in e} w(u) \geq 1$ for every hyperedge e , we have $X \cap e \neq \emptyset$, i.e. X is a hitting set. Conversely, every hitting set can be transformed into a solution of the Transversal ILP by giving weight one to the vertices of the hitting set and zero to the others. \square

Note that if the hypergraph contains an empty hyperedge, then the Transversal LP cannot be satisfied since there is a constraint $0 \geq 1$. Indeed, no vertex of the hypergraph can intersect the empty hyperedge. Let us now define the packing LP of a hypergraph $H = (V, E)$.

Packing (Integer) Linear Program:

Variables: A variable $x_e \in \mathbb{R}$ (or \mathbb{N}) for each hyperedge e .

Constraints: For every vertex v , $\sum_{e|v \in e} x_e \leq 1$.

For every hyperedge e , $x_e \geq 0$.

Objective function: maximize $\sum_{x_e \in E} x_e$.

Note that if the hypergraph H contains an empty hyperedge then the optimal value of the Packing Linear Program of H is infinite. Indeed, we can give to the empty hyperedge an arbitrarily large value since no constraint uses it.

Observation 2.14. *Let H be a hypergraph with no empty hyperedge. The optimal value of the Packing Integer Linear Program of H equals $\nu(H)$.*

Proof. First note that every variable has value 0 or 1. Indeed if a variable x_e has value at least two, then every vertex $v \in e$ would satisfy $\sum_{e' \mid v \in e'} x_e' \geq 2$, a contradiction. Such a vertex v exists since H does not contain any empty hyperedge.

So the Packing ILP can be seen as a weight function $w : E \rightarrow \{0, 1\}$. Let us denote by P the set of hyperedges e such that $w(e) = 1$. Since every vertex v satisfies $\sum_{e \mid v \in e} w(e) \leq 1$, every vertex is in at most one hyperedge of P . In other words the hyperedges of P are vertex disjoint, *i.e.* P is a packing. \square

If the hypergraph contains an empty hyperedge, then we can put it in the packing (it does not intersect any other hyperedge, so all the optimal packings contain it) and compute the packing of the remaining hypergraph via linear programming: containing an empty hyperedge does not avoid us to apply linear programming techniques.

Theorem 2.15. *Transversal Linear Program and Packing Linear Program are LP-dual.*

Sketch of proof. Let H be a hypergraph. Let us denote by A the matrix constraint, by b the constraint vector and by c the objective function of the Transversal LP. So the Transversal LP can be written as $\min^t cx$ under the constraint $Ax \leq b$.

In the dual linear program, there is a variable associated to each constraint of the primal linear program. In other words, there is a variable associated to every hyperedge constraint (we will denote it by x_e) and to every positivity constraint (we will denote it by y_v). In the following, we denote by y the concatenation of the vectors x_e and y_v . Let us first describe the objective function of the dual LP. Theorem 2.12 ensures that the objective function of the dual LP consists in maximizing $^t by$ (since the dual of a minimization problem is a maximization problem). The vector b equals one on the hyperedge constraints (since every hyperedge must have size at least one) and equals zero on the positivity constraints (since every variable must have non negative weight). So the objective function of the dual LP is $\max(\sum_{e \in E} x_e)$.

The vector c is a vector of ones (since we sum the values of all the variables in the objective function of the primal LP). The coefficients of the matrix A are 0 and 1. On the lines corresponding to hyperedge constraints, the coefficient $A(j, i) = 1$ if the vertex i is in the hyperedge e_j . On the lines corresponding to positivity constraints, all the coefficients equal zero except in the j -th column for the positivity constraint of the vertex v_j . The constraints of the dual LP correspond to the columns (*i.e.* to the vertices) of the primal LP. The dual constraint corresponding to the vertex v is $(\sum_{e \mid v \in e} x_e) + y_v = 1$. Theorem 2.12 ensures that the dual linear program has the following variables and constraints.

- Variables:** A variable x_e for each hyperedge e ,
- A variable y_v for every vertex v .
- Constraints:** For every vertex v , $y_v + \sum_{e \mid v \in e} x_e = 1$.
- For every hyperedge e , $x_e \geq 0$.
- For every vertex v , $y_v \geq 0$.

Note that each variable y_v appears in exactly one constraint (the constraint of the vertex v) and does not appear in the objective function. So we can forget these variables and replace the equalities

by inequalities. In other words, the vertex constraint for the vertex v becomes $\sum_{e|v \in e} x_e \leq 1$. It is equivalent to the original one since if we add y_v and give it the value $1 - \sum_{e|v \in e} x_e$, the original constraint is satisfied. The resulting LP is exactly the Packing LP. \square

Fractional transversality and integrality gap. In the following we will denote by τ^* and ν^* the fractional relaxation of respectively τ and ν . In other words, τ^* (resp. ν^*) is the optimal value of the Transversal (resp. Packing) Linear Program in real numbers. The following theorem is a direct consequence of Theorem 2.12 and Theorem 2.15.

Theorem 2.16. *Every hypergraph H with no empty hyperedge satisfies :*

$$\nu \leq \nu^* = \tau^* \leq \tau.$$

Before illustrating the integrality gap of transversal and packing LP, let us first provide a simple but useful observation.

Observation 2.17. *Let H be a hypergraph with no empty hyperedge and let c be a positive constant. If every hyperedge contains at least $c \cdot n$ vertices, then $\tau^* \leq 1/c$.*

Proof. Let w be the weight function $V \rightarrow \mathbb{R}$ such that $w(v) = 1/(cn)$ for every $v \in V$. Every hyperedge has weight at least 1 (since every hyperedge contains at least cn vertices and each vertex has weight $1/cn$), so all the constraints are satisfied. And the total weight of the vertex set is $n/cn = 1/c$. \square

Lemma 2.18. *The gap between τ and τ^* can be arbitrarily large.*

Proof. Let $\mathcal{U}_{n,2n}$ be the complete n -uniform hypergraph on $2n$ vertices. Since every hyperedge contains half of the vertices, Observation 2.17 ensures that $\tau^* \leq 2$. On the contrary, we have $\tau \geq n+1$. Otherwise the complement of a hitting set would have size at least n , and then would contain a hyperedge (since every subset of size n is a hyperedge), a contradiction. \square

Lemma 2.18 refines the fact that the gap between τ and ν can be arbitrarily large by Theorem 2.16.

Lemma 2.19. *The gap between ν and ν^* can be arbitrarily large.*

Proof. Let K_n be a clique on n vertices. Construct the following hypergraph H_n . The vertices of H_n are the edges of K_n . For every vertex v , create the hyperedge e_v containing all the edges adjacent to v . The hypergraph H_4 is represented on Figure 2.11.

Consider the following weight function w on the hyperedges which associates $1/2$ to every hyperedge of the hypergraph. The total weight is $n/2$ since there are n hyperedges in the hypergraph H_n and each hyperedge has weight $1/2$. The constraints are satisfied since every vertex of H_n is in two hyperedges (the edge uv of K_n is only in the hyperedges e_u and e_v). So $\nu^*(H_n) \geq n/2$. On the contrary, we have $\nu(H_n) = 1$. Indeed, for every pair of vertices u, v , the hyperedges e_u and e_v intersect on uv (the edge uv exists since the original graph is a clique). \square

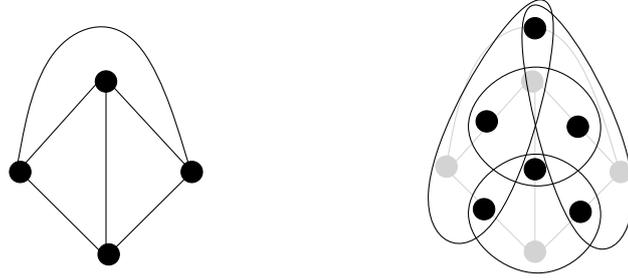


Figure 2.11: Illustration of the proof of Lemma 2.19. At the left a K_4 . At the right, the hypergraph constructed from the edges of K_4 .

Let us finally use both LP-duality and hypergraph duality in order to link α and c . Recall that α denotes the maximum size of a stable set of a hypergraph. And c denotes the minimum size of a covering of the hypergraph. Observation 2.9 ensures that $\tau(H) = c(H^t)$. And Observation 2.10 ensures that $\nu(H) = \alpha(H^t)$. Therefore, by applying Theorem 2.12 on the dual hypergraph, we have the following.

Observation 2.20. *Every hypergraph H satisfies $\alpha(H) \leq \alpha^*(H) = c^*(H) \leq c(H)$.*

2.2.3 Farkas' Lemma

Farkas

Lemma 2.21 (Farkas' Lemma). *Let A be a $n \times m$ matrix. At least one of the following holds:*

1. *There exists $w \in \mathbb{R}^m$ such that $w \geq 0$, $w \neq 0$ and $Aw \geq 0$.*
- or 2. *There exists $y \in \mathbb{R}^n$ such that $y \geq 0$, $y \neq 0$ and ${}^t y A \leq 0$.*

There exists several proof of this result. In particular it can be seen as a particular case of the well-known Hahn-Banach theorem (which is a main theorem in analytics). There also exists less involved proofs (one of them can be found in [180]). Note that Lemma 2.21 is one of the most famous “visual” applications of the duality of linear programming. Indeed it ensures that either the primal satisfies some property or that the dual satisfies the dual property. Let us now give a nice application of Farkas' Lemma due to Alon et Brightwell [6].

Lemma 2.22 (Alon, Brightwell [6]). *For every oriented graph $D = (V, A)$, there exists a weight function $w : V \rightarrow [0, 1]$ such that $w(V) = 1$ and for each vertex u , $w(N^+(u)) \geq w(N^-(u))$.*

Proof. Let M be the adjacency matrix of the oriented graph D , that is to say that $M_{u,v} = 1$ if $uv \in A$, -1 if $vu \in A$, and 0 otherwise. Apply Lemma 2.21 to M .

If case one occurs, then w is a nonnegative weight function on the columns of M , with at least one non zero weight. Since $Mw \geq 0$, we get $w(N^+(u)) \geq w(N^-(u))$ for all $u \in V$. Indeed, each constraint corresponds to each line of the matrix. And the line corresponding to the vertex u gives the constraint $\sum_{v|uv \in A} w(v) - \sum_{v|vu \in A} w(v) \geq 0$, i.e. we have $w(N^+(u)) \geq w(N^-(u))$. We conclude by rescaling the weight function with a factor $1/w(V)$ (such a rescaling does not affect the inequalities and provides a total weight of 1).

Otherwise, case two occurs and there is $y \in \mathbb{R}^n$ with $y \neq 0$ such that ${}^t y A \leq 0$. We get by transposition

${}^tAy \leq 0$ thus $-Ay \leq 0$ since A is an antisymmetric matrix, and then $Ay \geq 0$. We conclude as in the previous case. \square

Note that the inequality of Lemma 2.22 can also be reversed. In other words, there exists a weight function $w : V \rightarrow [0, 1]$ such that $w(V) = 1$ and for each vertex u , $w(N^-(u)) \geq w(N^+(u))$. Indeed, if we reverse the arcs of an oriented graph and if we apply Lemma 2.22 on the oriented graph with reversed edges, then it immediately gives $w(N^-(u)) \geq w(N^+(u))$. The following Lemma is a direct consequence of Lemma 2.22

Lemma 2.23 (Alon, Brightwell [6]). *Every tournament T has a weight function $w : V \rightarrow [0, 1]$ such that $w(V) = 2$ and for each vertex u , $w(N^+[u]) \geq 1$.*

Proof. Let us consider a weight function w' satisfying conditions of Lemma 2.22. Let us define the weight function w such that for every $u \in V$ we have $w(u) = 2w'(u)$. The total weight is 2, and for every vertex u we have $w(N^+(u)) \geq w(N^-(u))$. In addition, since T is a tournament, we have $w(N^+[u]) + w(N^-(u)) = 2$. So Lemma 2.22 ensures that $w(N^+(u)) \geq 1$. It immediately gives the conclusion. \square

2.3 Algorithmic aspects of hitting sets. Application to MULTICUT

All along this part we will consider parameterized algorithmic aspects of hitting sets. First we will introduce the HITTING SET problem. In Section 2.3.1, we will focus more precisely on graph separation problems and MULTICUT. We will make a state of the art of the existing results in this area. Then we will introduce two crucial tools to design parameterized algorithms for graph separation problems. The first one, introduced by Marx, is called the important separator technique. The second one, due to Marx and Razgon, consists in a random sampling of important separators. In Sections 2.3.2 and 2.3.4 we will present these tools. The proofs of these two sections are quite involved compared to the other parts of this chapter.

Let us first state formally the decision hitting set problem.

HITTING SET:

Input: A hypergraph $H = (V, E)$, an integer k .

Parameter: k .

Output: TRUE if H has a hitting set of size at most k , otherwise FALSE.

A lot of graph problems can be expressed as hitting set problems. The easiest one is VERTEX COVER. Indeed, as underlined in Section 2.1, a vertex cover of the graph G is a hitting set of a 2-uniform hypergraph G . So HITTING SET is NP-complete since VERTEX COVER is (it is one of the 21 Karp's problem). Note also that DOMINATING SET is a HITTING SET problem. Indeed let G be a graph and consider its closed neighborhood hypergraph H of G . A hitting set of H is a dominating set of G . Indeed a hitting set of H is a subset of vertices intersecting every closed neighborhood, so a set of vertices whose closed neighborhood cover the vertices of G . Hence a hitting set of H is a dominating set of G . So, from a parameterized point of view, HITTING SET is $W[2]$ -hard parameterized by the size of the solution because DOMINATING SET is (and the reduction does not modify the parameter), see [75]. Finally HITTING SET (probably) does not admit FPT algorithms in the general case.

In this manuscript, we are interested in a particular type of HITTING SET problems, which are the graph separation problems. Let us first provide several definitions. Given a graph G and a set R of requests between pairs of vertices (these vertices are called *terminals* or *endpoints*), a *multicut*¹ is a subset F of edges of G whose removal separates the two endpoints of every request (*i.e.* for every request, the two endpoints of this request lie in different connected components of $G \setminus F$).

requests
terminals
multicut

MULTICUT:

Input: A graph $G = (V, E)$, a set of requests R , an integer k .

Parameter: k .

Output: TRUE if there is a multicut of size at most k , otherwise FALSE.

Consider the hypergraph H on vertex set E where $E' \subseteq E$ is a hyperedge if the set of edges of E' form a path between two endpoints of a same request. In other words, the hyperedges are the possible paths between endpoints of requests. A hitting set of H is a subset of edges which intersects all the paths between endpoints of the same requests. So the deletion of such a subset of edges put the endpoints of every request in distinct connected components. In other words, a hitting set is a multicut of (G, R) .

MULTICUT parameterized by the size of the solution was considered as one of the main open problems of the fixed parameterized complexity theory [69]. We proved with Jean Daligault and Stéphan Thomassé that MULTICUT is FPT [33]. Independently Marx and Razgon proved the same result in [147]. The rest of this section is organized as follows. In Section 2.3.1, we propose a state of the art of the MULTICUT problem and related problems. Most of the recent results obtained on graph separation problems are based on a technique called important separators and its randomized extension called shadow removal. In Section 2.3.2, we present the important separators technique and illustrate it on MULTIWAY CUT. Finally in Section 2.3.4 we introduce the shadow removal technique and apply it to DIRECTED MULTIWAY CUT.

2.3.1 Multicut problems

MULTICUT and its variants have raised an extensive literature. These problems play an important role in network issues, such as routing and telecommunications (see [62]). For example, vertices of the graph could represent Urban Switch Centers in a telephone network, and (weighted) edges represent physical connections between vertices [43].

The MULTICUT problem is already hard when the input graph is a tree since VERTEX COVER can be viewed as MULTICUT in stars. Indeed consider a star on n branches where each branch corresponds to a vertex of the graph. There is a request between two branches if the corresponding vertices are adjacent in the graph. One can easily verify that a multicut of size at most k is a vertex cover of the original graph of size at most k (and conversely). Hence MULTICUT IN TREES is NP-complete and MaxSNP hard (since VERTEX COVER is), which implies that it admits no Polynomial-Time Approximation Scheme (PTAS) unless $P=NP$. Garg *et al.* [104] proved that MULTICUT IN TREES admits a 2-approximation algorithm, by showing that in trees the minimal size of a multicut is at most twice the maximal flow value, and using a primal-dual approach. Guo and Niedermeier [112] proved that MULTICUT IN TREES is FPT parameterized by the size of the solution. The existence of a polynomial kernel was asked in [22]. We answered this with Jean Daligault, Stéphan Thomassé and Anders Ye

1. In this thesis, the term multicut stands for edge-multicut.

in [34] by proving that MULTICUT IN TREES admits a $\mathcal{O}(k^6)$ kernel. This upper bound was improved by Chen *et al.* in [51] into a $\mathcal{O}(k^3)$ kernel.

A classical variant of MULTICUT is the MULTIWAY CUT problem in which a set of terminals has to be pairwise separated.

MULTIWAY CUT:

Input: A graph $G = (V, E)$, a set of terminals $X \subseteq V$, an integer k .

Parameter: k .

Output: TRUE if there exist k edges whose deletion puts every vertex of X in distinct connected components, otherwise FALSE.

MULTIWAY CUT has been proved to be FPT parameterized by the size of the solution by Marx [143]. A faster $O^*(4^k)$ algorithm was proposed by Chen *et al.* [52], improved into a $\mathcal{O}^*(2^k)$ algorithm in [66]. The proof, based on important separators, will be detailed a little bit further. Recently Kratsch and Wahlström proved that MULTIWAY CUT admits a $\mathcal{O}(k^t)$ kernel where t denotes the number of terminals. It is still open to determine if MULTIWAY CUT admits a polynomial kernel parameterized by the size of the cutset only.

On general instances, Garg *et al.* gave an approximation algorithm for MULTICUT within a logarithmic factor in [103], proving that the minimum size of the multicut is within a factor $\mathcal{O}(\log(\ell))$ of the maximum multifold value in general graphs, where ℓ is the number of requests. However, MULTICUT has no constant factor approximation algorithm if Khot's Unique Games Conjecture holds [48].

Guo *et al.* showed in [111] that MULTICUT is FPT when parameterized by both the treewidth of the graph and the number of requests. Gottlob and Lee proved a stronger result in [106]: MULTICUT is FPT when parameterized by the treewidth of the input structure, *i.e.* the input graph whose edge set is augmented by the set of requests. Marx proved that MULTICUT is FPT parameterized by the size of the solution plus the number of requests [143]. A faster algorithm running in time $\mathcal{O}^*((8 \cdot \ell)^k)$ was given by Guillemot [110] (recall that ℓ is the number of requests). Marx *et al.* [144] obtained FPT results for more general types of constrained MULTICUT problems through treewidth reduction results. However their treewidth reduction techniques do not yield FPT algorithm of MULTICUT when parameterized by the size of the solution only. Finally, Marx and Razgon obtained a factor 2 Fixed-Parameter-Approximation for MULTICUT parameterized by the size of the solution in [146]. We finally proved in 2011 with Jean Daligault and Stéphan Thomassé that MULTICUT is FPT parameterized by the size of the solution [33]. Independently, Marx and Razgon also provided an FPT algorithm [147]. Nevertheless there remain several open problems on MULTICUT. In particular we can add constraints on the structure of the target MULTICUT: we can look for independent multicuts, or connected multicuts. When parameterized by both the size of a solution and the number of requests, the problem is FPT for finding independent multicuts [145]. Recently, Cygan *et al.* proved that both MULTICUT and MULTIWAY CUT parameterized by the size of the solution do not admit polynomial kernels [64].

Problem 1. Does MULTICUT parameterized by both the size of the solution and the number of requests admits a polynomial kernel?

Instead of wanting to intersect all the paths between every pair of terminal of a same request,

we can just ask for intersecting a subset of paths between every pair of terminals. More formally, consider the following problem.

HITTING PATH:

Input: A graph G , a set R of paths in G , an integer k .

Parameter: k .

Output: TRUE if there is a set of at most k edges of G which hits R , otherwise FALSE.

Problem 2. *Is HITTING PATH FPT parameterized by the size of the solution?*

Note that HITTING PATH and MULTICUT are rather distinct problems: MULTICUT cannot be (easily) reduced to HITTING PATH in polynomial time since there can be an exponential number of paths between some endpoints of a request.

Recently, lots of research have been done for studying MULTICUT in directed graphs. The input graph is a directed graph and the requests are directed pairs. The objective is to eliminate all the directed paths between directed pairs of terminals. Marx and Razgon proved that DIRECTED MULTICUT is $W[1]$ -hard parameterized by the size of the solution in [147]. Even worse, Kratsch *et al.* proved that DIRECTED MULTICUT is $W[1]$ -hard parameterized by the size of the solution in directed acyclic graphs [130]. Nevertheless DIRECTED MULTICUT in Directed Acyclic Graphs is FPT parameterized by the size of the solution plus the number of pairs of terminals [130]. In addition, Chitnis *et al.* proved in [55] that DIRECTED MULTICUT with two pairs of terminals is FPT parameterized by the size of the solution.

Problem 3. *Is DIRECTED MULTICUT FPT parameterized by both the size of the solution and the number of terminals?*

Or, a little bit weaker, is DIRECTED MULTICUT FPT parameterized by the size of the solution when the number of requests is a fixed constant?

Nevertheless in several sub-cases, DIRECTED MULTICUT is FPT. A crucial variant of DIRECTED MULTICUT is FPT parameterized by the size of the solution in directed graphs: SKEW MULTICUT. In the SKEW MULTICUT problem, we are given an oriented graph and two ordered sets of terminals s_i, t_i and we want to find a subset of at most k edges which eliminates all the directed paths from s_i to t_j for every $j \geq i$. SKEW MULTICUT is FPT parameterized by the size of the solution in Directed Acyclic Graphs [53]. The proof of this result is the core of the proof that DIRECTED FEEDBACK VERTEX SET is FPT parameterized by the size of the solution [53]. More recently and using shadow removal techniques, several other variants of DIRECTED MULTICUT such as DIRECTED MULTIWAY CUT were shown to be FPT [55].

2.3.2 Important separators

All along the manuscript, the graphs are assumed to be simple and loopless. Nevertheless the rest of this chapter and exclusively there, we will assume that the graphs can have multiple edges but still no loop. In other words, for every pair u, v there can be several edges uv in the graph. Let us now introduce important separators and shadow removal. These two techniques are very involved. Therefore the proofs of the next sections are much more complicated than the other proofs of this

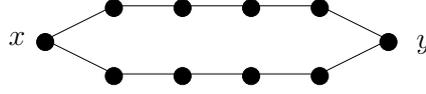


Figure 2.12: Every pair of edges with one edge in the upper path and one edge in the below path is a minimum xy -separator.

chapter. We will nevertheless detail the first proofs in order to increase little by little the complexity. We will also give at the beginning the main intuitions on important separators.

Lots of proofs of graph separation problems solved in the last few years are based on important separators, such as [33, 53, 55, 143, 147]. This technique has been introduced by Marx in [143]. All the lemmas and theorems of Section 2.3.2 are classical ones, and were already proved in [52, 143] for instance.

Let G be a graph and x, y be two vertices of a graph. The number of minimum separators between x and y can be arbitrarily large. Consider for instance the graph of Figure 2.12. Every pair of edges with one edge on the above path and one on the below path is a minimum separator between x and y . So, if the two xy -paths have length n , then there are 2^n xy -separators of size 2. Nevertheless, all of them are not necessarily “important”. Imagine for instance that our goal is to separate x from y in such a way the component of x is minimized. Then there is a unique such xy -separator of minimum size (instead of an exponential number). In Figure 2.12 it is the set of two edges adjacent to x . Marx proved that the number of (indivisible) important xy -separators of size at most k can be bounded by a function of k . In addition, all these separators can be found in FPT time parameterized by k .

Let us now define formally these notions. Let $G = (V, E)$ be a connected graph on n vertices with a particular vertex x called the *root*. In the following, we will only consider separators from the point of view of the root x . A separator can be considered as a set of edges, but also as a bipartition of the vertex set (the vertices which are in the connected component of x and the vertices which are not). In the following we consider separators as bipartitions. As we want to focus on one side of the bipartition, we define a *separator* as a subset of vertices S containing the root x . An xy -separator is a subset of vertices containing x and not containing y . The *border* of S is the set of edges of G with exactly one endpoint in S . We denote the border of S by $\Delta(S)$, and the cardinality of $\Delta(S)$ is denoted by $\delta(S)$. By abuse of notations, the *size of a separator* S will denote the size of the border of S , i.e. $\delta(S)$. Recall that \bar{A} denotes the complement of the set A , i.e. $V \setminus A$.

Observation 2.24. *The function δ is submodular, i.e. for every pair of separators S, T we have $\delta(S) + \delta(T) \geq \delta(S \cap T) + \delta(S \cup T)$.*

Proof. Let us prove that every edge which contributes to the right hand side also contributes to the left hand side with at least the same multiplicity. Let S and T be two separators and let u, v be two vertices of the graph.

- If uv is in $\Delta(S \cap T)$ and in $\Delta(S \cup T)$, then without loss of generality, u is in $S \cap T$ and v in $\bar{S} \cap \bar{T}$. So uv is in both $\Delta(S)$ and $\Delta(T)$.
- If uv is in $\Delta(S \cap T)$ and not in $\Delta(S \cup T)$, then, up to symmetry, we have $u \in S \cap T$ and $v \in S \cap \bar{T}$. So uv is in $\Delta(T)$.

- If uv is in $\Delta(S \cup T)$ and not in $\Delta(S \cap T)$ then, without loss of generality, u is in $S \setminus (S \cap T)$ and v is in $\overline{S \cap T}$. So in particular uv is in $\Delta(S)$.

□

Let x be a root. Let y be a vertex. The tools used for proving Theorem 2.5 (Menger's Theorem) ensures that the size of a minimum xy -separator can be determined in polynomial time (and a minimum separator can be computed in polynomial time) and it is equal to the maximum number of edge-disjoint paths between x and y . The *connectivity* between x and y is the minimum size of a separator between x and y and it is denoted by $\lambda(x, y)$ (or when no confusion is possible by λ). A separator S is an *important separator* if every separator T such that $T \subsetneq S$ satisfies $\delta(T) > \delta(S)$. Let us first make an easy observation.

connectivity

important separator

Observation 2.25. *Every separator S contains an important separator S' such that $\delta(S') \leq \delta(S)$.*

Proof. We prove it by induction on the number of vertices contained in S . Either S is an important separator and the conclusion holds. Otherwise there exists $S' \subsetneq S$ such that $\delta(S') \leq \delta(S)$. By induction there exists an important separator S'' such that $S'' \subset S'$. We also have $S'' \subseteq S$. □

The “idea” of important separators can be summarized as follows. If you want to find a separator with less vertices (in the component of the root x), then you have to pay a price: the size of the border of the separator must be strictly larger. Before stating the most important results of this section, let us first give some general properties of important separators.

Lemma 2.26. *Important separators are closed under union. In other words, if S and T are important separators, then $S \cup T$ is an important separator.*

Proof. Let $S_1 \cup S_2$ be the union of two important separators. Let $S_3 \subsetneq S_1 \cup S_2$ be a separator such that $\delta(S_3)$ is minimized. Our goal is to prove that $\delta(S_3) > \delta(S_1 \cup S_2)$. Indeed in this case, since $\delta(S_3)$ minimizes the size of the border over the strict subsets of $S_1 \cup S_2$, it would mean that no $T \subsetneq S_1 \cup S_2$ satisfies $\delta(T) \leq \delta(S_1 \cup S_2)$, i.e. $S_1 \cup S_2$ is an important separator.

Without loss of generality, we can assume that S_1 is not included in S_3 . Since S_1 is an important separator, we have $\delta(S_1 \cap S_3) > \delta(S_1)$. Indeed $S_1 \cap S_3 \subsetneq S_1$ and every strict subset T of S_1 satisfies $\delta(T) > \delta(S_1)$ since S_1 is an important separator. As Observation 2.24 ensures that $\delta(S_1 \cap S_3) + \delta(S_1 \cup S_3) \leq \delta(S_1) + \delta(S_3)$, we obtain $\delta(S_1 \cup S_3) < \delta(S_3)$.

Note that we have $S_1 \cup S_3 \subseteq S_1 \cup S_2$ (since both S_1 and S_3 are in this set). By definition, S_3 has minimum border among strict subsets of $S_1 \cup S_2$ and $\delta(S_3) > \delta(S_1 \cup S_2)$, hence the set $S_1 \cup S_3$ is not a strict subset of $S_1 \cup S_2$. So it is equal to $S_1 \cup S_2$. Finally $\delta(S_1 \cup S_2) = \delta(S_1 \cup S_3) < \delta(S_3)$, thus $S_1 \cup S_2$ is an important separator. □

Lemma 2.27. *If S_1, S_2 are distinct important separators, then $\delta(S_1 \cup S_2) < \max(\delta(S_1), \delta(S_2))$.*

Proof. As $S_1 \neq S_1 \cup S_2$ or $S_2 \neq S_1 \cup S_2$, we can assume without loss of generality that $S_1 \subsetneq S_1 \cup S_2$. By Lemma 2.26, $S_1 \cup S_2$ is an important separator and $S_1 \subsetneq S_1 \cup S_2$, thus we have $\delta(S_1 \cup S_2) < \delta(S_1) \leq \max(\delta(S_1), \delta(S_2))$. □

A separator S is an *indivisible separator* if no strict subset of $\Delta(S)$ is a separator. In other words, when we delete from the graph all the edges of $\Delta(S)$ but exactly one, the graph becomes a connected

indivisible separator

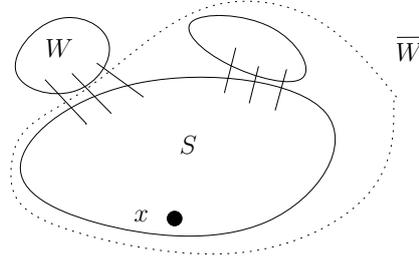


Figure 2.13: Illustration of Lemma 2.28.

graph. Or yet differently an indivisible separator is a separator S such that $V \setminus S$ induces a connected subgraph. A separator is *divisible* if it is not indivisible. An *indivisible important xy -separator* S is an indivisible important separator rooted in x such that $y \notin S$.

Lemma 2.28. *If S is a divisible important separator and W is a connected component of $G \setminus S$, the separator \overline{W} is an indivisible important separator with $\delta(\overline{W}) < \delta(S)$.*

Proof. The situation of Lemma 2.28 is illustrated on Figure 2.13. The separator \overline{W} is indivisible since the set W is connected. So we just have to prove that \overline{W} is an important separator. First note that we have $\Delta(\overline{W}) \subsetneq \Delta(S)$. Note that the non equality comes from the fact that S is divisible.

Consider an important separator $T \subseteq \overline{W}$ which minimizes $\delta(T)$. Our goal is to prove that $T = \overline{W}$. By Lemma 2.26, $S \cup T$ is an important separator. As $\delta(T) \leq \delta(S \cup T)$ by minimality of $\delta(T)$ (since $S \cup T \subset \overline{W}$), we have $T = S \cup T$ (because $S \cup T$ is an important separator). In particular, $S \subseteq T$. Every edge of $\Delta(\overline{W})$ has one endpoint in S and one endpoint in W . Since $S \subseteq T$ and since no vertex of W is in T , we have $\Delta(\overline{W}) \subseteq \Delta(T)$. Therefore, by minimality of $\delta(T)$, we have $\Delta(T) = \Delta(\overline{W})$. Finally we have $T = \overline{W}$. Thus, \overline{W} is an important separator. \square

Corollary 2.29. *Every indivisible separator S contains an indivisible important separator S' with $\delta(S') \leq \delta(S)$.*

Proof. Let S'' be an (non necessarily indivisible) important separator contained in S such that $\delta(S'') \leq \delta(S)$. As \overline{S} is connected, the set \overline{S} is included in a connected component Y of $G \setminus S''$. By Lemma 2.28, $S' := \overline{Y}$ is an indivisible important separator with $\delta(S') < \delta(S'') \leq \delta(S)$. Moreover, $S' \subseteq S$ as $\overline{S} \subseteq Y$. \square

In the following we are looking for xy -separators: we do not care about separating x from other vertices than y , so the only separators which are interesting for us are indivisible xy -separators since if the separator is divisible we can do “the same” with less edges. So Corollary 2.29 ensures that we can just look for indivisible important xy -separators. Marx first proved that the number of indivisible important xy -separators of size at most k is bounded by a function of k . In the literature the “indivisible” is often omitted and such indivisible important xy -separators are often called important xy -separators.

Lemma 2.30. *Let G be a graph and x, y be two vertices. There is a unique important xy -separator of size $\lambda(x, y)$ (which can be computed in polynomial time). This separator is indivisible. In addition, every important xy -separator is contained in S .*

Proof. Let S be an important xy -separator of size $\lambda(x, y)$. Note that such a separator exists by Observation 2.25. Assume by contradiction that an important xy -separator T contains a vertex which is not in S . Lemma 2.26 ensures that $S \cup T$ is an important separator. Since $S \subsetneq S \cup T$ and since $S \cup T$ is an important separator, we have $\delta(S \cup T) < \delta(S)$. In addition $y \notin S \cup T$, so there is a xy -separator of size less than $\lambda(x, y)$, a contradiction. So no important xy -separator contains a vertex which is not in S .

So there is a unique minimum important xy -separator. In addition this separator is indivisible otherwise it would not be minimal. The proof of the complexity part can be found in [143] for instance. \square

In the following the unique important xy -separator of size $\lambda(x, y)$ will be called *the minimum important xy -separator*. Let us now give some definitions. Let $G = (V, E)$ be a graph. The graph $G' = (V', E')$ obtained by *identifying* two vertices u and v is the graph where u, v are deleted and are replaced by a unique vertex w , and w is adjacent to every vertex w' which is a neighbor of u or v and the multiplicity of this edge is the multiplicity of uw' plus the multiplicity of vw' . *Contracting* an edge e consists in identifying the endpoints of the edge e . Let us make an observation which will be important in the proof of the next theorem.

Observation 2.31. *Let G be a graph and S be the minimum important xy -separator. Let G' be the graph G where all the vertices of \bar{S} have been identified with y . Every indivisible important xy -separator of G is an indivisible important xy -separator of G' .*

Proof. Lemma 2.30 ensures that, if we look for important xy -separators, the vertices which are not in the unique minimum indivisible important separator S can be identified with y . Indeed for every indivisible important xy -separator T , no vertex of \bar{S} is in T . So if we identify all the vertices of \bar{S} (and still denote by y the identified vertex), the indivisible important xy -separators are not affected. \square

Note also that in the graph G' there is a unique xy -separator of size $\lambda(x, y)$ (which is $V \setminus y$). Note that we just deal with separators and not important separators here. The first upper bound on the number of indivisible important xy -separators is due to Marx. It was improved by Chen *et al.* We will see further that this bound is nearly tight.

Theorem 2.32 (Marx [143], Chen *et al.* [52]). *Let G be a graph and x, y be two vertices. There are at most 4^k indivisible important xy -separators of size at most k which can be enumerated in FPT time (in $4^{\mathcal{O}(k)}$).*

Proof. First note that if $\lambda(x, y) > k$, then there is no indivisible important xy -separator. So in the following we assume that $k \geq \lambda(x, y)$. By Lemma 2.30, there exists a unique minimum indivisible important xy -separator S . Identify all the vertices in \bar{S} with y (since no vertex of \bar{S} is in an indivisible important xy -separator by Lemma 2.30 this operation is safe). We still denote by y the identified vertex. Let us denote by G' the graph obtained after this contraction. Observation 2.31 ensures that after this contraction:

- There is a unique minimum xy -separator which is the set of edges adjacent to y .
- The indivisible important xy -separators of the graph G are exactly the indivisible xy -separators of the graph G' .

In the rest of the proof we only work on the graph G' . Let us prove that every indivisible important xy -separator can be found using a branching algorithm. Let e be an edge of $\Delta(S)$.

Claim 2.33. *Every indivisible important xy -separator of G' containing e in its border is an indivisible important xy -separator of $G' - e$ (i.e. the graph obtained by a deletion of e from E).*

Proof. Let T be an indivisible important xy -separator such that $e \in \Delta(T)$. Assume by contradiction that $T \setminus e$ is not important in $G' - e$. Then there exists $T' \subsetneq T$ such that $\delta(T') \leq \delta(T)$. In the whole graph G' (i.e. when we add e), the border of T' increases by at most one (since only the edge e can be added in the border of T'). And the border of T increases by exactly one. So we still have $\delta(T') \leq \delta(T)$ in G' , contradicting the fact that T is an important xy -separator. In addition $T \setminus e$ is clearly still indivisible in $G' - e$. \square

Claim 2.34. *Every indivisible important xy -separator of G' not containing e in its border is an indivisible important xy -separator of the graph G' where the endpoints of e are contracted.*

Proof. Let us denote by G'_e the graph G' in which the edge e is contracted. Let T be an indivisible important xy -separator of G' such that $e \notin \Delta(T)$. So both endpoints of e are in the same connected component in $G[E \setminus \Delta(T)]$ (since e is not in $\Delta(T)$). Assume by contradiction that T is not indivisible in G'_e . So there exists $T' \subsetneq T$ such that $\delta(T') \leq \delta(T)$. Though the size of $\delta(T')$ does not increase in the graph G' . So we still have $\delta(T') \leq \delta(T)$ and $T' \subsetneq T$, a contradiction. \square

Consider the following algorithm. If $\lambda(x, y) > k$ return the empty set. Otherwise, compute the unique minimum indivisible important xy -separator S (which can be done in polynomial time by Lemma 2.30). Identify all the vertices of \bar{S} with y . Let e be an edge adjacent to y in the contracted graph. Claim 2.33 and 2.34 ensures that if we branch by deleting the edge e or by contracting the edge e , then we find all the indivisible important xy -separators.

Let us prove that in both cases, the invariant $2k - \lambda$ decreases. In the first case, k decreases by one and the connectivity decreases by exactly one. Indeed Theorem 2.5 ensures that there are $\lambda(x, y)$ edge-disjoint xy -paths in G , so there remain at least $\lambda - 1$ paths when e is deleted, i.e. the connectivity decreases by at most one (and actually exactly one). In the second case, k is not modified and λ strictly increases. Indeed since vertices of \bar{S} have been contracted with y and since S is the unique minimum important xy -separator, the contraction of the edge e ensures that the connectivity strictly increases. Otherwise there would be an xy -separator T of size λ with both vertices of e in \bar{T} , and then $T \subsetneq S$. So the invariant decreases. So the height of the branching tree is at most $2k$. And the width of the branching tree is 2 since we branch over two possible choices at each step. So there are at most 4^k branches and each branch gives at most an indivisible important xy -separator, so there are at most 4^k indivisible important xy -separators of size at most k . Since computing the unique minimum indivisible important xy -separator can be done in polynomial time, the resulting algorithm is FPT. \square

Theorem 2.32 ensures that the number of indivisible important xy -separators is bounded. We have already seen that the “important” is necessary in Figure 2.12. But the “indivisible” assumption is also necessary. Indeed in Figure 2.14, any pair of edges containing the edge xy is an important xy -separators while the number of indivisible important xy -separators is one. The next lemma ensures that the upper bound of Theorem 2.32 is almost tight.

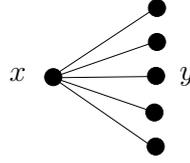


Figure 2.14: Every pair of edges containing the edge xy is an important xy -separator, but there is a unique indivisible important xy -separator which is the edge xy .

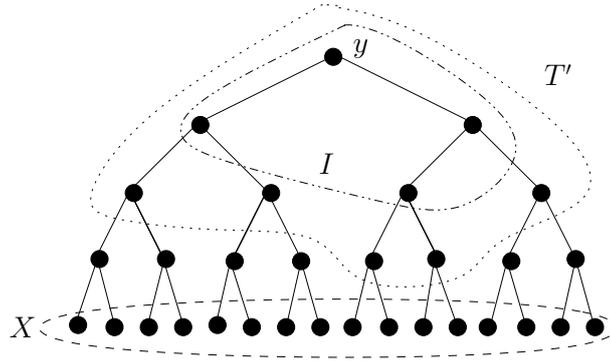


Figure 2.15: A graph with almost $\mathcal{O}^*(4^k)$ indivisible important YX -separators of size k . The sets T' and I are illustrations of the proof of Lemma 2.35.

Before stating formally the lemma, let us give some definitions. Let X, Y be two subsets of vertices. In the following, we deal with separators rooted in X and with XY -separators. A separator rooted in X is a separator which contains all the vertices of X . In other words, if we identify all the vertices of X into x , it is a separator rooted in x . A XY -separator is a separator which does not contain any vertex of Y . The notion of important separators can be naturally extended to XY -separators.

Lemma 2.35 (Chen *et al.* [52]). *Let X, Y be two sets of vertices. There exist graphs with $\mathcal{O}(4^k / \text{poly}(k))$ indivisible important YX -separators of size k .*

Proof. Let us first give some definitions. A *binary tree* (rooted in y) is a tree with nodes of degree one or three except the root y which has degree two. A complete binary tree (rooted in y) of depth k is a tree in which every node of depth at most $(k - 1)$ has degree three and every node of depth k is a leaf. Let us first recall two facts on binary trees. First a binary tree with ℓ leaves has exactly $(\ell - 1)$ internal nodes (an internal node is a node which is not a leaf). Second the number of binary trees rooted in y with exactly ℓ leaves is the Catalan number $C_{\ell-1}$ where

$$C_{\ell-1} = \frac{1}{\ell} \binom{\ell-1}{2\ell-2} \sim 4^\ell / \text{poly}(\ell).$$

Let T be the complete binary tree of depth k . Let us denote by y its root and by X the set of leaves of T . Let us prove that T has $4^k / \text{Poly}(k)$ indivisible important XY -separators of size at most k (note that trees are rooted in y while separators are rooted in X). The tree T is illustrated in Figure 2.15.

Claim 2.36. *Let T' be a binary subtree of T with ℓ leaves. Let I be the set of internal nodes of T' . The set \bar{I} is an indivisible important Xy -separator with $\delta(\bar{I}) = \ell$.*

Proof. First note that \bar{I} is a Xy -separator since $y \in I$. It is indivisible since every vertex of \bar{I} is by construction in the connected component of a vertex of X . And we have $\delta(I) = \delta(\bar{I}) = \ell$ since the set of internal nodes of a binary tree is only adjacent to its set of leaves. Let us finally prove that \bar{I} is important. Assume by contradiction that there exists $S \subsetneq \bar{I}$ such that $\delta(S) \leq \delta(\bar{I})$. Since \bar{I} is indivisible, Corollary 2.29 ensures that we can assume that S is indivisible. So \bar{S} is connected, *i.e.* induces a subtree of T rooted in y . Let L' be the set of vertices adjacent to a vertex of \bar{S} . The set $\bar{S} \cup L'$ induces a binary tree. Since $|\bar{S}|$ is strictly larger than $|I|$, the short observations ensures that we have $|L'| > \ell$. A contradiction with $\delta(S) \leq \delta(I)$. \square

Let us finally prove that every binary tree T' with exactly k leaves appears as a subgraph of T (rooted in y). Since T' has k leaves, it has $(k-1)$ internal nodes. Hence the depth of T' is at most k . Since T is the complete binary tree of depth k , it contains T' as a subgraph.

So there are at least C_{k-1} indivisible important YX -separators, which achieves the proof of Lemma 2.35. \square

Finding the exact upper bound of the number of indivisible important separators is still open. Note nevertheless that this question is much more a combinatorial question than an algorithmic question since finding the exact number of indivisible important separators will not improve the complexity of algorithms since the current bound is nearly tight (4^k as upper bound and $4^k / \text{poly}(k)$ as lower bound).

Problem 4. *Let x, y be two vertices. What is the maximum number of indivisible important xy -separators of size at most k ?*

2.3.3 Application to MULTIWAY CUT

The first application of important separators in parameterized complexity has been proposed by Marx for the MULTIWAY CUT problem. Let us first recall the formal definition of the MULTIWAY CUT problem.

MULTIWAY CUT:

Input: A graph $G = (V, E)$, a set $T \subseteq V$ of terminals, an integer k .

Parameter: k .

Output: TRUE if there exists k edges whose deletion puts vertices of T in pairwise distinct connected components, otherwise FALSE.

Lemma 2.37 (Pushing Lemma [143]). *Let (G, T, k) be an instance of MULTIWAY CUT. Let $x \in T$. If the instance is positive, then there exists a solution containing an indivisible important $(T \setminus x, x)$ -separator.*

Proof. The idea is the following: if we separate a terminal x from the other terminals then it is better to take the maximum number (by inclusion) of vertices in the connected component of x . Indeed, it will be easier to separate the remaining terminals in the remaining graph since it contains less vertices. Let us denote by Y the set $T \setminus x$.

Let F be a solution of a MULTIWAY CUT instance of size at most k . Let F' be a minimum (by inclusion) subset of F such that F' is a Yx -separator. The set F' is the border of an indivisible Yx -separator and we denote by A the Yx -separator (rooted in Y) of border F' . By Lemma 2.29, there exists an indivisible important Yx -separator B such that $B \subseteq A$ and $\delta(B) \leq |F'|$.

Let us prove that $(F \setminus \Delta(A)) \cup \Delta(B)$ is also a solution of the instance (G, T, k) of the MULTIWAY CUT problem. The size of $(F \setminus \Delta(A)) \cup \Delta(B)$ is at most k since $\delta(B) \leq \delta(A)$. Let P be a path between two vertices of Y in B . Then it is a path between two vertices of Y in A since $B \subseteq A$. Since F is a multiwaycut, the path P is intersected by an edge of F . This edge is not in $\Delta(A)$ by definition of the path, so it is in $F \setminus \Delta(A)$, *i.e.* in $(F \setminus \Delta(A)) \cup \Delta(B)$. So every path P between vertices of Y are hit by $(F \setminus \Delta(A)) \cup \Delta(B)$. Since B is a Yx -separator, the paths between the vertex x and a vertex of Y are also hit. Hence $(F \setminus \Delta(A)) \cup \Delta(B)$ is a multiwaycut satisfying the conditions of the pushing lemma. \square

Corollary 2.38 (Marx [143]). *MULTIWAY CUT is FPT parameterized by the size of the solution.*

Sketch of the proof. Let us prove it by induction on the number of terminals. Let (G, T, k) be an instance of the MULTIWAY CUT problem. If there is a unique terminal, then return true. Otherwise, let $x \in T$ and Y be the set $T \setminus x$. If $\lambda(x, Y) > k$ then return false. If there is no Yx -path, delete x from T since x is already separated from the other terminals.

In the other cases, Theorem 2.32 ensures that there are at most 4^k indivisible important Yx -separators of size at most k which can be found in FPT time. Lemma 2.37 ensures that if there exists a solution, then there exists a solution such that the border of one of these separators is in the solution. Branch over all the indivisible important Yx -separators. In other words, for every indivisible important Yx -separator A , delete the edges of $\Delta(A)$ in the graph G , decrease by $\delta(A)$ the size of the target solution and remove x from T . Lemma 2.37 ensures that the initial instance is positive if and only if one of the branches is positive. Since each important Yx -separator has size at least 1 (since there exist Yx -paths), the parameter k strictly decreases at each branching step. So the branching width is 4^k and the branching depth is k : the resulting algorithm is FPT. \square

Actually, a more careful analysis gives a $\mathcal{O}^*(4^k)$ algorithm [52]. We will use the important separator technique for proving that MULTICUT is FPT in Chapter 6. In their proof, Marx and Razgon introduced a new technique, called shadow removal [147].

2.3.4 Shadow removal technique

In Sections 2.3.4 and 2.3.5, and exclusively in these ones, we will deal with vertex graph separation problems instead of edge graph separation problems. In other words instead of deleting edges, we delete vertices of the graph. The notion of important separators can be easily translated for vertex separators (and the statements for edge-separators still hold for vertex separators). Let $G = (V, E)$ be a graph and $x \in V$. A vertex *separator* (rooted in x) is a subset S of vertices which contains x . The border of S , denoted by $\Delta(S)$ if the set of vertices $z \notin S$ such that z a neighbor in S . We denote by $\delta(S)$ the size of $\Delta(S)$. A (vertex) *important separator* is a subset S of vertices such that for every $S' \subsetneq S$ we have $\delta(S') > \delta(S)$. An xy -separator is *indivisible* if all the vertices of $\Delta(S)$ are needed to separate x from y . Note that it does not mean (as for edge-separators) that \bar{S} is connected. The shadow removal technique has been introduced by Marx and Razgon in [147] and has

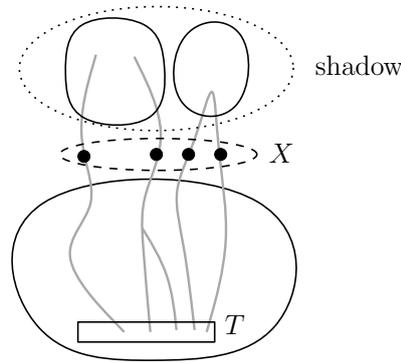


Figure 2.16: The shadow of X (which is a \mathcal{F} -HITTING SET of the the gray structure) is the above part of the figure: it is the set of vertices which cannot reach T .

been extensively used since [55, 130, 136, 137]. This method is based on a random sampling of important separators.

Let $G = (V, E)$ be a graph and T be a set of terminals. Let \mathcal{F} be a subset of connected subgraphs of G such that every $F \in \mathcal{F}$ satisfies $F \cap T \neq \emptyset$. Such a set \mathcal{F} of subgraphs is called a T -compatible set. Given a T -compatible set \mathcal{F} , the \mathcal{F} -hypergraph of G is the hypergraph on vertex set V with hyperedge set \mathcal{F} . The \mathcal{F} -HITTING SET problem is defined as follows:

\mathcal{F} -HITTING SET:

Input: A graph $G = (V, E)$, a set T of terminals, a T -compatible set \mathcal{F} , an integer k .

Output: TRUE if the \mathcal{F} -hypergraph of G satisfies $\tau \leq k$, otherwise FALSE.

In this section, we consider that separators are rooted in T . In Figure 2.16, the set \mathcal{F} composed of the three gray connected subgraphs is a T -compatible subset. The set X is a hitting set of the \mathcal{F} -hypergraph. Note that MULTICUT is an \mathcal{F} -HITTING SET problem where the elements of \mathcal{F} are all the paths between pairwise disjoint terminals. Every path of \mathcal{F} induces a connected subgraph which intersects T . And a hitting set of \mathcal{F} is a multicut as underlined in Section 2.3.1.

Let X be a \mathcal{F} -HITTING SET. The *shadow* of X is the set of vertices which are not reachable from T in $G[E \setminus X]$. In other words, the shadow of X is the set of vertices which are not in the connected component of any vertex of T in $G[V \setminus X]$. Roughly speaking the shadow of X is the set of vertices which are hidden by X when the vertices of T are “enlightened”. In Figure 2.16, the shadow of X is the above part of the figure. Note that, the shadow of an \mathcal{F} -hitting set can be empty (in this case all the vertices are in the connected component of a vertex of T). A hitting set X of the \mathcal{F} -hypergraph without vertex in the shadow of X is called a *shadowless solution*. Let us first make an observation on the structure of the \mathcal{F} -hitting sets.

Observation 2.39. *Let G be a graph, T a set of terminals and \mathcal{F} a T -compatible set and S, S' two separators such that $S' \subseteq S$. If $\Delta(S)$ is an \mathcal{F} -hitting set, then $\Delta(S')$ is an \mathcal{F} -hitting set.*

In particular if S is a \mathcal{F} -hitting set then the border of any important separator contained in S is also a \mathcal{F} -hitting set.

Proof. Let $F \in \mathcal{F}$. Since $\Delta(S)$ is an \mathcal{F} -hitting set, there exists a vertex s of $\Delta(S)$ such that $s \in F$. If $s \in \Delta(S')$ then F is also hit by $\Delta(S')$. Otherwise, since $S' \subseteq S$, the vertex s is not in S' . Since $T \cap F \neq \emptyset$

T -
compatible
set
 \mathcal{F} -
hypergraph

shadow

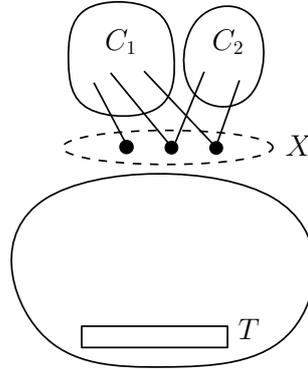


Figure 2.17: The component C_1 is in the exact shadow of X while C_2 is not since X is an indivisible TC_1 -separator but a divisible TC_2 -separator.

and F is connected, there exists a path in G from T to s . This path intersects $\Delta(S')$, so F is hit by $\Delta(S')$. \square

The following theorem is the key result of the shadow removal technique.

Theorem 2.40 (Marx, Razgon [147]). *Let G be a graph, T a set of terminals and \mathcal{F} a T -compatible set. In $\mathcal{O}^*(2^{\mathcal{O}(k)})$, we can compute a set Z such that, if the \mathcal{F} -hypergraph has a hitting set of size at most k , then with probability $2^{-\mathcal{O}(k)}$ the \mathcal{F} -hypergraph has a minimum hitting set X such that:*

- The set Z contains the shadow of X .
- No vertex of X is contained in Z .

shadow re-
moval

Sketch of the proof. The proof is based on a procedure that randomly samples borders of indivisible important Tv -separators for every vertex v and take as set Z the union of their shadows. Let us prove that we have to take and to avoid a bounded number (in k) of important separators during the sample in order to satisfy both points. Before formalizing this procedure with Algorithm 2, let us introduce a new notion. The *exact shadow* of a set $X \subseteq V \setminus T$ is the set of vertices v of $V \setminus (T \cup X)$ such that X is the border of a minimal (by inclusion) Tv -separator. In other words, X is the border of an indivisible Tv -separator. Figure 2.17 illustrates exact shadows. Again differently, a connected component of $G[V \setminus X]$ is nice for X if X is exactly the set of vertices which are not in X with an edge with an endpoint of X . The exact shadow of X is the union of all the connected components which are nice for x . Note that the exact shadow of X could be empty.

exact
shadow

Let us prove that Algorithm 2 satisfies the condition of Theorem 2.40 (except that the probability will not be as good as expected). First note that for every $v \in V$, the set of indivisible important Tv -separators can be enumerated in $4^{\mathcal{O}(k)}$ time by Theorem 2.32. So the time complexity of Algorithm 2 satisfies the conditions of Theorem 2.40. For proving Theorem 2.40, we need the following claim.

Claim 2.41. *Let S be an important separator set and let $X = \Delta(S)$. Every vertex $v \notin S \cup X$ is in the exact shadow of some $X_v \subseteq X$.*

Hint of proof. We just have to prove that for every connected component C of $G[V \setminus (S \cup X)]$ then $\Delta(C) \cap X$ is an indivisible important TC -separator. The scheme of the proof looks like the proof of

Algorithm 2: Shadow removal algorithm**Input** : A graph $G = (V, E)$, a set $T \subseteq V$, a T -compatible set \mathcal{F} , an integer k .**Output** : A set Z satisfying the conditions of Theorem 2.40.

- 1 For every vertex v , compute all the indivisible important Tv -separators of size at most k (if an important separator is computed several times, keep it only once).
- 2 Sample them with probability one half.
- 3 Compute Z which is the union of the exact shadows of the borders of the important separators selected during the sample.
- 4 Return Z .

Lemma 2.37. Indeed if we can put less vertices in the connected component of T and still separate T and C then we would find a set $S' \subsetneq S$ such that $\delta(S') \leq \delta(S)$, a contradiction since S is an important separator. \square

Let us now prove that Algorithm 2 is correct. If the \mathcal{F} -hypergraph has no hitting set of size at most k , there is nothing to prove. So we can assume that there is a \mathcal{F} -hitting set of size at most k . Let X be such a solution. By Observation 2.39, we can assume that X is the border of an important separator S . Let Y be the shadow of X . Claim 2.41 ensures that for every vertex y of Y , there exists $X_y \subseteq X$ such that X_y is an indivisible important Ty -separator. Since each separator is taken with probability one half and since there are 2^k subsets of X , then all the vertices of Y are in the set Z computed by Algorithm 2 with probability at least 2^{-2^k} . So the first point holds.

Let us prove the second point. If a vertex x of X is in the set Z computed by Algorithm 2 then it is in the exact shadow of an important separator sampled during the algorithm. But every vertex is in the exact shadow of at most 4^k important separators (which are actually the indivisible important xT -separators). Since there are at most k vertices in X , vertices of X are in the exact shadow of at most $k \cdot 4^k$ indivisible important separators, so none of them is chosen with probability at least $2^{-2^{\theta(k)}}$.

Note that this proof does not provide a probability better than $2^{-2^{\theta(k)}}$ (instead of $2^{-\iota(k)}$). But using a better distribution on the set of important separators, this algorithm can be improved in order to obtain the desired probability. \square

Note that, Algorithm 2 does not need to know \mathcal{F} for finding Z . Indeed during the execution of the algorithm, we just compute indivisible important separators and randomly sample them and we do not use the structure of \mathcal{F} at all. We just use the structure on \mathcal{F} for proving the correctness. In particular, it means that even if there is an exponential number of elements in \mathcal{F} , the algorithm is only exponential in k .

Also note that taking exact shadows and not shadows in the proof of Theorem 2.40 is necessary. Indeed consider Figure 2.18. Assume that the vertex x must not appear in the set Z computed by Algorithm 2. The vertex x is in the exact shadow of one indivisible important Tx -separator which has border the unique neighbor of x ; so with probability one half it does not appear in Z . On the contrary, assume that the algorithm makes the union of the shadows instead of the exact shadows. Then for every vertex y of Y , there is an indivisible important Ty -separator containing the neighbor

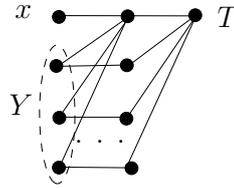


Figure 2.18: The vertex x is in the exact shadow of one important separator but on the shadow of an arbitrarily large number of important separators.

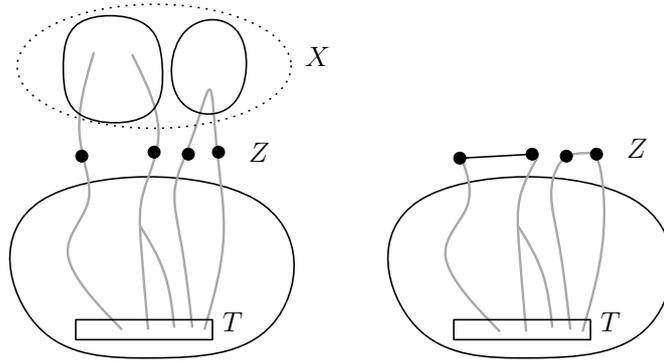


Figure 2.19: The modification of the graph G into G' during the proof of Lemma 2.42.

of x . So if the set Y has size n , the probability that x is in the (non exact) shadow of a selected invisible important separator is $1 - (1/2)^n$.

Theorem 2.40 ensures that we can, with positive probability, transform an instance into an instance which has a shadowless solution. Let us state it more formally and prove it.

Lemma 2.42. *Let $I = (G, T, \mathcal{F}, k)$ be an instance of \mathcal{F} -HITTING SET where G is a graph, T a set of terminals and \mathcal{F} a T -compatible set. In $\mathcal{O}^*(2^{\mathcal{O}(k)})$, we can compute an instance $I' = (G', T, \mathcal{F}', k)$ such that:*

- Every \mathcal{F}' -hitting set of I' is a \mathcal{F} -hitting set of I .
- If I has a \mathcal{F} -hitting set of size at most k then I' has a shadowless \mathcal{F}' -hitting set of size at most k with probability $2^{-\mathcal{O}(k)}$.

Proof. The proof consists in an application of Theorem 2.40. Let Z be a set of vertices computed by the algorithm of Theorem 2.40. Let us first construct an instance I' from I . Next we will prove that I' satisfies both conditions of Lemma 2.42. Construct the following graph G' with vertex set $V \setminus Z$. Every edge of $G[V \setminus Z]$ is an edge of G' . In addition, for every pair of vertices x, y of $G[V \setminus Z]$, if there exists a xy -path with interior vertices in Z then add the edge xy in the graph G' . In other words, if x and y can be connected by a path in Z then we "simulate" this path by an edge. The instance I' is the instance (G', T, \mathcal{F}', k) where every hyperedge F of \mathcal{F} becomes $F \cap (V \setminus Z)$ in \mathcal{F}' . The modification of the graph G into G' is illustrated in Figure 2.19. Let us prove that any hyperedge F' in the \mathcal{F}' still induces a connected subgraph of G' . Assume by contradiction that F' has at least two connected components. Let F_1 be a connected component and $F_2 = F' \setminus F_1$. Since F is connected in G , there

exists a path whose interior vertices are in Z connecting a vertex of F_1 with a vertex of F_2 . So in G' , there is an edge between a vertex of F_1 and a vertex of F_2 , a contradiction.

If the set X is a hitting set of the \mathcal{F}' -hypergraph of G' then it is a hitting set of the \mathcal{F} -hypergraph of G . Indeed every hyperedge of \mathcal{F} contains a hyperedge of \mathcal{F}' . So the first point holds.

Let us now prove the second point. Let X be a \mathcal{F} -hitting set of size at most k for the instance I . First note that the shadow of X in G' is included in the shadow of X in G . Indeed assume that $x \in V \setminus Z$ in not in the shadow of X in G . Then there exists a path P from a vertex of T to x in $V \setminus X$. By replacing every subpath of vertices of Z by an edge of G' (which exists by construction), the same path exists in G' . So x is not in the shadow of X in G' .

Let us denote by Y the shadow of X . Let Z be the set computed by Theorem 2.40. Theorem 2.40 ensures that with probability $2^{-\mathcal{O}(k)}$, the set Z contains all the vertices of Y and does not contain any vertex of X . So with probability $2^{-\mathcal{O}(k)}$ the set X is included in $V \setminus Z$. Hence X is in the vertex set of G' . Since X intersects every set of \mathcal{F} in G , the set X still intersects every elements of \mathcal{F} in G' by construction of G' . Thus X is a \mathcal{F} -hitting set of the instance $I' = (G', T, \mathcal{F})$. Finally with probability $2^{-\mathcal{O}(k)}$ the solution X is a shadowless. Indeed with probability $2^{-\mathcal{O}(k)}$ all the vertices of the shadow of X in G are deleted in G' and every vertex in the shadow of X in G' is in the shadow of X in G . \square

So, the shadow removal technique consists in transforming an instance into another where we have to look for a shadowless solution (with positive probability). Hence, up to application of Theorem 2.40, we can assume that all the vertices of $G[V \setminus X]$ are in a connected component of a vertex of T : this method has been used intensively since the introduction of this technique. Finally the shadow removal technique can be derandomized.

Theorem 2.43 (Marx, Razgon [147]). *The algorithm of Theorem 2.40 can be derandomized in FPT time parameterized by k .*

2.3.5 Application to DIRECTED MULTIWAY CUT

The shadow removal technique was first used for proving that Multicut is FPT in [147], but since this application is quite involved, we prefer illustrate this method on DIRECTED MULTIWAY CUT. Let us first introduce the notion of shadow in directed graphs. Let D be a directed graph and T be a set of terminals. Let \mathcal{F} be a subset of subgraphs of G such that for every $F \in \mathcal{F}$, every vertex v of F can reach T and can be reached from T in the directed graph induced by the vertices of F (this condition replaces the connectivity condition on undirected graphs). Such a set of subgraphs is called a T -compatible set. The \mathcal{F} -hypergraph of G is the hypergraph on vertex set V with hyperedge set \mathcal{F} . The shadow of X is the set of vertices which are not reachable from T in $G[E \setminus X]$ or which cannot reach T in $G[E \setminus X]$. The following statement is the pendant of Theorem 2.40 for directed graphs.

Theorem 2.44 (Chitnis *et al.* [55]). *Let D be a directed graph, T a set of terminals, \mathcal{F} a T -compatible set, an integer k . In $\mathcal{O}^*(f(k))$, we can compute a set Z such that, if there exists a \mathcal{F} -hitting set of size at most k , then with probability $2^{-\mathcal{O}(k^2)}$ there is a minimum \mathcal{F} -hitting set X such that:*

- The set Z contains the shadow of X .
- No vertex of X is contained in Z .

Note that the probability that the “good” event occurs is $2^{-\mathcal{O}(k^2)}$ in directed graphs which is not as good as the probability $2^{-\mathcal{O}(k)}$ obtained for undirected graphs. As for undirected graphs, the

algorithm of Theorem 2.44 can be derandomized. Let us finally prove that DIRECTED MULTIWAY CUT is FPT parameterized by the size of the solution using Theorem 2.44.

Theorem 2.45 (Chitnis *et al.* [55]). *DIRECTED MULTIWAY CUT is FPT parameterized by the size of the solution.*

Proof. Let T be the set of terminals of a DIRECTED MULTIWAY CUT instance (D, T, k) . The set \mathcal{F} corresponds to all the directed paths between distinct pairs of terminals. Note that these directed paths are T -compatible since the endpoints of the paths are in T and every vertex can be reached by the beginning of the path and can reach the end of the path. Following the scheme of the proof of Lemma 2.42 we can prove the following for directed graphs.

Claim 2.46. *Let $I = (D, T, \mathcal{F}, k)$ be an instance of the \mathcal{F} -HITTING SET problem where D is a directed graph, T a set of terminals and \mathcal{F} a T -compatible set. In $\mathcal{O}^*(f(k))$, we can compute a new instance $I' = (D', T, \mathcal{F}', k)$ such that:*

- *Every \mathcal{F} -hitting set of I' is a \mathcal{F} -hitting set of $I = (D, T, \mathcal{F}, k)$.*
- *If I has a \mathcal{F} -hitting set of size at most k then I' has a shadowless \mathcal{F} -hitting set of size at most k with probability $2^{-\mathcal{O}(k^2)}$.*

In the following, we consider the instance I' which is the triple (D', T, \mathcal{F}', k) of the reduced instance of Claim 2.46. Let us denote by V' the set of vertices of D' . Assume that there exists a \mathcal{F} -hitting set X of size at most k in D' . Since X is shadowless, for every vertex $v \notin X$, there exists a path from v to a vertex of T in $D'[V' \setminus X]$ and a path from a vertex of T to v in $D'[V' \setminus X]$. So there exists a directed path P_1 from $t_1 \in T$ to v and a directed path P_2 from v to $t_2 \in T$ in the graph $D'[V' \setminus X]$. So we have $t_1 = t_2$ since otherwise there would remain a directed path between two distinct vertices of T which is not intersected by X , a contradiction since X is a directed multiway cut. So after the deletion of X , all the vertices are in the strong connected component of a vertex of T . So the vertex set $V' \setminus X$ can be partitioned into $|T|$ sets such that each set is the strong connected component of a vertex of T . Note that there is no arc between distinct components, otherwise there would be a directed path between distinct vertices of T . So X is a multiway cut of the underlying graph of D' . Conversely let G' be the underlying graph of D' . A multiwaycut of G' is a directed multiway cut of D' . Indeed if every non-oriented path is hit, then every oriented path also is. So G' has a multiway cut of size k if and only if D' has a multiway cut of size k . Since MULTIWAY CUT is FPT parameterized by k by Corollary 2.38, DIRECTED MULTIWAY CUT also is. \square

VC-dimension

This chapter is devoted to giving the definitions and theorems related to VC-dimension. We give applications of VC-dimension to graph theory all along this chapter. I have participated to the proofs of two of these applications:

- *A dichotomy result for identifying codes in Section 3.2.3 which is joint work with Zhentao Li, Aurélie Lagoutte, Aline Parreau and Stéphan Thomassé.*
- *A proof of Scott's conjecture for maximal triangle-free graphs in Section 3.4.2 which is joint work with Stéphan Thomassé.*

This chapter is split into several parts. In Section 3.1, we formally define shattered sets and VC-dimension, then we study the VC-dimension and its stability via opposite, complement and dual operations. All along this chapter, various examples will be provided in order to illustrate the notions. Section 3.2 is devoted to proving that hypergraphs of bounded VC-dimension admit a polynomial number of hyperedges [178]. In addition, the provided upper bound is tight. We will propose several proofs of this key lemma and give a dichotomy theorem for identifying codes. In Section 3.3, we will state the most relevant result on VC-dimension which ensures that the integrality gap between τ and τ^* cannot be arbitrarily large when the VC-dimension is bounded [117]. The proof is based on the fact that the hypergraph admits a polynomial number of hyperedges. Recall that Lemma 2.18 ensures that the gap between τ and τ^* can be arbitrarily large for general hypergraphs. We illustrate the interest of this statement with a proof of Alon et al. [6] on dominating set in k -majority tournaments and on the chromatic number of dense graphs [140]. Another application will be provided in Chapter 5. However, the gap between τ and ν can be arbitrarily large for hypergraphs of bounded VC-dimension. Section 3.4 will introduce two stronger versions of VC-dimension which ensure the Erdős-Pósa property. The first one, called 2VC-dimension, provides a polynomial gap between τ and ν [71]. We will illustrate this result with a particular case of the so-called Scott's conjecture [41]. The other one, due to Matoušek [151], ensures that both (p, q) -property and bounded VC-dimension implies Erdős-Pósa property. A result of Chepoi et al. [54], generalized in Chapter 4, will illustrate this result.

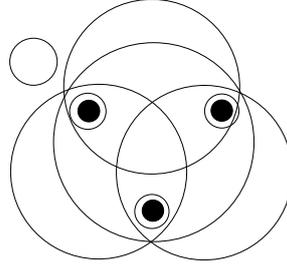


Figure 3.1: A shattered set of size 3.

3.1 Shattered sets

shattered

Let H be a hypergraph. A subset X of vertices is *shattered* if for every subset $X' \subseteq X$, there exists a hyperedge e such that $e \cap X = X'$. Note that the empty set is always shattered. Figure 3.1 represents a shattered set of size 3. In other words, a shattered set is a subset X of vertices such that all the possible traces of H on X exist. Yet differently, the hyperedges intersect in all the possible ways the set X . In the following, we denote by $sh(H)$ the number of shattered sets in H . Since a shattered set is a subset of vertices, we have $sh(H) \leq 2^n$.

A shattered set is a witness of the local complexity of the hypergraph: a hypergraph is complex on a set X if all the traces on X exist. The interest of shattered sets (and of VC-dimension) is that a bounded local complexity provides several general properties on the hypergraph, for instance on the number of hyperedges (Theorem 3.12) or on the size of hitting sets (Theorem 3.20 and Theorem 3.32).

VC-dimension

The *Vapnik-Chervonenkis dimension*, or *VC-dimension* for short, is the maximum size of a shattered set. In the following we will only deal with hypergraphs of VC-dimension at least one. Indeed as long as a hypergraph contains two (distinct) hyperedges, it contains a shattered set of size one. A hypergraph has VC-dimension at least d if and only if it contains \mathcal{C}_d as a subhypergraph. Let us denote by $vc(H)$ the VC-dimension of H . Given a family of hypergraphs \mathcal{H} , the VC-dimension of \mathcal{H} is the largest VC-dimension of a hypergraph of \mathcal{H} . The VC-dimension was first introduced by Vapnik and Chervonenkis in 1971 in [190].

The VC-dimension has been used in various areas such as learnability theory [21, 92, 116] (where VC-dimension is one of the main concepts), statistics [189], extremal graph and hypergraph theory [71, 100, 140, 154], combinatorial geometry and discrepancy [124, 125, 148, 160] and, more recently, graph theory [6, 41, 54, 140]. In this thesis we deal with graph theoretical problems.

3.1.1 Opening properties

Observation 3.1. Every hypergraph H of VC-dimension d satisfies $sh(H) \leq \sum_{i=0}^d \binom{n}{i}$.

Proof. The proof is straightforward: if the VC-dimension is at most d , then only sets of size at most d can be shattered. And the number of such sets is $\sum_{i=0}^d \binom{n}{i}$. \square

Note that the number of shattered sets gives more precise information than the VC-dimension. Indeed, Observation 3.1 ensures that as long as the VC-dimension is at most d , the number of shat-

tered sets is at most $\sum_{i=0}^d \binom{n}{i}$. Nevertheless, a reverse function which does not depend on n does not exist.

Observation 3.2. *For every n , there exist hypergraphs with n vertices and at most n hyperedges and at most n shattered sets with VC-dimension $\lfloor \log n \rfloor$.*

Proof. Let V be a set of size n and X be a subset of V of size $\lfloor \log n \rfloor$. Denote by H the hypergraph with vertex set V whose hyperedges are all possible subsets of X . By construction, the shattered sets are the subsets of X . Indeed, since no hyperedge contains a vertex which is not in X , no set containing a vertex which is not in X is shattered. Finally, $vc(H) = \lfloor \log n \rfloor$ (since X is shattered) but $sh(H) = 2^{\lfloor \log n \rfloor} \leq n$. \square

Note that the VC-dimension of a hypergraph $H = (V, E)$ is at most $\lfloor \ln |E| \rfloor$. Indeed, consider a shattered set X . All the possible traces exist on X . Since there are $2^{|X|}$ traces on every set X (all the possible subsets of X), the hypergraph contains at least $2^{|X|}$ hyperedges.

In the rest of this section, we study the variation of the VC-dimension for the complement, opposite and dual operations. We will see that their behaviors (from a VC-dimension point of view) are various but nevertheless interesting. First, the VC-dimension of the complement hypergraph is the same as the original hypergraph. In the case of the opposite hypergraph, another notion of shattering naturally appears, called strong shattering. For dual hypergraphs, we will illustrate the notion of dual shattered set with complete Venn diagrams, and we will look in particular for the largest gap between the VC-dimension and the dual VC-dimension.

Complement hypergraph. Recall that the complement hypergraph H^c is the hypergraph whose hyperedges are $V \setminus e$ for every $e \in E$.

Observation 3.3. *Every hypergraph H satisfies $vc(H^c) = vc(H)$.*

Proof. Let X be a shattered set of H . Let X' be a subset of X . Since X is shattered in H , there exists a hyperedge e whose trace on X is $X \setminus X'$. So the trace on X of the complement hyperedge e is X' , i.e. a hyperedge of H^c has trace X' on X . So X is shattered in H^c . \square

This proof underlines that the crucial aspect in the VC-dimension theory is not the hyperedge e itself but the pair of hyperedges (e, \bar{e}) . So replacing all the hyperedges by their complements does not affect the VC-dimension. However the size of the hyperedges provides an upper bound on the VC-dimension. More precisely, we have the following.

Observation 3.4. *Every hypergraph containing only hyperedges of size at most k has VC-dimension at most k .*

Proof. If a set X is shattered, then a hyperedge e satisfies $e \cap X = X$. So, we have $|X| \leq k$. \square

Observation 3.4 ensures that k -uniform hypergraphs have VC-dimension at most k . Let us determine the VC-dimension of complete uniform hypergraphs.

Observation 3.5. *The complete uniform hypergraph $\mathcal{U}_{k,n}$ has VC-dimension $\min(k, n - k)$.*

Proof. First assume that $k \leq n/2$. Let X be a subset of vertices of size k . Since $k \leq n/2$, every subset X' of X can be completed into a set Y of size k with vertices of $V \setminus X$. By definition, Y is a hyperedge of $\mathcal{U}_{k,n}$ whose trace on X is X' . So X is shattered, and the VC-dimension of $\mathcal{U}_{k,n}$ is at least k . Observation 3.4 ensures that the VC-dimension of $\mathcal{U}_{k,n}$ is at most k , which provides the equality.

If $k \geq n/2$ then we use Observation 3.3 and compute the VC-dimension of the complement hypergraph of $\mathcal{U}_{k,n}$. Since the complement of $\mathcal{U}_{k,n}$ is $\mathcal{U}_{n-k,n}$ and since $n-k \leq n/2$, the first part of the proof ensures that the VC-dimension of $\mathcal{U}_{k,n}$ equals $n-k$. \square

Opposite hypergraph. Recall that the opposite hypergraph of H is the hypergraph H^o with vertex set V where e is a hyperedge of H^o if and only if e is not a hyperedge of H . A subset X of vertices is *strongly shattered* if there exists a subset $Y \subseteq V \setminus X$ such that for every subset $X' \subseteq X$, the hyperedge $Y \cup X'$ exists. In other words, a strongly shattered set is a subset of vertices such that all its subsets can be extended into a hyperedge in the same way in the whole hypergraph. Note that the difference between shattered and strongly shattered set is a permutation of the order of the quantifiers. Indeed a set is shattered if “for every subset of X , there exists a subset of $V \setminus X$ ” and a set is strongly shattered if “there exists a subset of $V \setminus X$ such that for every subset of X ”. In the following $ssh(H)$ will denote the number of strongly shattered sets in H . Note that $ssh(H) \leq sh(H)$ since every strongly shattered set is shattered.

Observation 3.6. *Let H be a hypergraph. For every subset X of vertices of H , either X is shattered in H or $V \setminus X$ is strongly shattered in H^o .*

Proof. If X is not shattered, then there is a subset $X' \subseteq X$ such that no hyperedge of H satisfies $X \cap e = X'$. In other words, the set X' cannot be extended into a hyperedge with vertices of $V \setminus X$. So, in the opposite hypergraph, it means that every subset of $V \setminus X$ can be extended into a hyperedge with X' . Hence the set $V \setminus X$ is strongly shattered in the opposite hypergraph.

If X is shattered, then for every subset X' of X , there exists a subset Y of $V \setminus X$ such that $X' \cup Y$ is a hyperedge. So, in the opposite hypergraph it means that $V \setminus X$ cannot be strongly shattered using the set X' . And since $V \setminus X$ can only be extended with subsets of X , $V \setminus X$ is not strongly shattered in H^o . \square

The interest of the link between shattered sets and strongly shattered sets will be illustrated in Section 3.2.

Dual hypergraph. Recall that the dual hypergraph of H is the hypergraph where hyperedges become vertices, and vertices become hyperedges. The *dual VC-dimension* of H is the VC-dimension of the dual hypergraph of H . There is another way to define the dual VC-dimension which is more intuitive. A subset of edges $E' = \{e_1, \dots, e_\ell\}$ forms a *complete Venn diagram* if for every subset E'' of E' there exists a vertex x such that for every $e \in E'$, $x \in e$ if and only if $e \in E''$ (see Figure 3.2 for an illustration). In other words, there is at least one vertex in each possible intersection. The *dual VC-dimension* is also the maximum size of a complete Venn diagram.

The two definitions are equivalent. Indeed, consider a complete Venn diagram. Each hyperedge becomes a vertex in the dual hypergraph. Let E'' be a subset of E' and x be a vertex such that for every $e \in E'$, $x \in e$ if and only if $e \in E''$. Then the hyperedge corresponding to x in the dual

strongly
shattered

dual VC-
dimension

complete
Venn
diagram

dual VC-
dimension

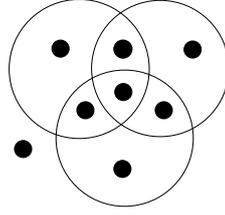


Figure 3.2: A complete Venn diagram with 3 hyperedges.

hypergraph contains all the vertices of E'' and does not contain those of $E' \setminus E''$. Therefore the set E' is shattered in the dual hypergraph. One can easily verify that a shattered set in the dual hypergraph is a complete Venn diagram in H .

The following result, due to Assad [16], ensures that the gap between the VC-dimension and the dual VC-dimension cannot be arbitrarily large.

Lemma 3.7 (Assouad [16]). *Every hypergraph H of VC-dimension d has dual VC-dimension at least $\lfloor \log d \rfloor$.*

Proof. Assume that the VC-dimension of H is at least 2^d . Let us prove that H admits a complete Venn diagram of size d . Let $X = \{x_1, \dots, x_{2^d}\}$ be a shattered set. There exists a hyperedge e_1 containing $x_1, \dots, x_{2^{d-1}}$ and not containing the others vertices of X . It induces an equal bipartition of the set X denoted by X_1, X_2 (where each set has size 2^{d-1}). Since X is shattered, there exists e_2 containing exactly one half of both X_1 and X_2 . We repeat this operation as long as each subset contains at least two vertices. And the end of the procedure, we obtain d hyperedges $E' = \{e_1, \dots, e_d\}$ since the original set has size 2^d and each subset is divided by 2 equal subsets at each step.

Let us prove that E' is a complete Venn diagram, *i.e.* let us verify that, for every subset I of $\{1, \dots, d\}$, there is a vertex in $E'' = (e_i)_{i \in I}$ and not in $E' \setminus E''$. To do so, let us prove by induction that exactly 2^{d-i} vertices of X satisfy the intersection constraints on $\{e_1, \dots, e_i\}$. The result holds for $i = 0$. Assume that there is a subset Y of X of size 2^{d-i} which satisfies the intersection constraints on $\{e_1, \dots, e_i\}$. By construction, e_{i+1} separates the set Y into two equal parts, one included in e_{i+1} and the other disjoint with e_{i+1} . So exactly one half of Y satisfies the constraint on e_{i+1} . So 2^{d-i-1} vertices satisfy the intersection constraints on $\{e_1, \dots, e_{i+1}\}$.

After d steps, there remains exactly one vertex. So there is a complete Venn diagram of size d , which achieves the proof of Lemma 3.7. \square

The gap provided by Lemma 3.7 is tight. Indeed consider a hypergraph on n vertices containing all the possible hyperedges. Such a hypergraph has VC-dimension n (since each subset of vertices is a hyperedge). Its dual hypergraph contains exactly n edges, so a shattered set has size at most $\lfloor \ln(n) \rfloor$ since $vc(H) \leq \lfloor \ln(E) \rfloor$.

Note that $vc((H^d)^d) = vc(H)$ since the dual of the dual hypergraph is the initial hypergraph. Hence, Lemma 3.7 ensures that the dual VC-dimension is at most $2^{d+1} - 1$. More precisely, we have:

Lemma 3.8. *Every hypergraph H satisfies:*

$$\lfloor \log(vc(H)) \rfloor \leq vc(H^d) \leq 2^{vc(H)+1} - 1.$$

This exponential gap can be reduced into a polynomial gap in many cases. Consider for instance auto-dual hypergraphs, *i.e.* hypergraphs such that H^d is isomorphic to H . For such hypergraphs, we have $vc(H) = vc(H^d)$. Neighborhood hypergraphs are auto-dual hypergraphs. For such hypergraphs, the vertices are the vertices of a graph G and Y is a hyperedge if there exists a vertex v of G such that $N_G(v) = Y$. Such hypergraphs are auto-dual since we have $x \in N_G(y)$ if and only if $y \in N_G(x)$. These types of hypergraphs will be studied in Chapter 4.

Bipartite graphs. Recall that a hypergraph can be seen as its bipartite incident graph $B = ((V, E), E')$ where ve is an edge of E' if $v \in e$ in the hypergraph. A set X of vertices of V of size d is shattered if in the bipartite graph B , for every $X' \subseteq X$ there exist a vertex $e \in E$ such that $N_B(e) \cap X = X'$. In other words, the neighborhoods of vertices of E provide all the possible traces on the set X . In other words, for every subset X' of X , there exists a vertex e of E such that $N(e) \cap X = X'$ in the incidence bipartite graph.

Given a (bipartite) graph $G = ((V, W), E)$, the VC-dimension of G from V to W is the VC-dimension of the neighborhood hypergraph from V to W . Note that the VC-dimension of a hypergraph $H = (V, E)$ is exactly the VC-dimension of the bipartite incidence graph from E to V .

Lemma 3.9. *Every bipartite graph $((V, W), E)$ of VC-dimension $2d$ from V to W contains an induced copy of every bipartite graph of size $d \times d$.*

Proof. Let $G = B((V, W), E)$ be a bipartite graph of VC-dimension at least $2d$. Let $G' = B((A, B), F)$ be a bipartite graph of size $d \times d$. Let us prove that an induced subgraph of G is isomorphic to G' . We denote by a_1, \dots, a_d the vertices of A and by b_1, \dots, b_d the vertices of B . Let $X = \{x_1, \dots, x_{2d}\}$ be a shattered set of size $2d$ in G . Denote by X_1 the set $\{x_1, \dots, x_d\}$. Our goal is to construct a bijection such that the vertex x_i of X_1 is isomorphic to a_i in A of G' .

For every vertex b_i of B , denote by X_{b_i} the subset of X_1 such that $x_j \in X_{b_i}$ if and only if $a_j \in N(b_i)$. Since X is shattered, there exists a vertex y_i of W such that $N(y_i) = X_{b_i} \cup \{x_{d+i}\}$. Note that the vertex x_{d+i} is a neighbor of y_i and is not a neighbor of any other vertex y_j with $j \neq i$. In particular, for every i, j , we have $y_i \neq y_j$. Let us denote by Y the set $\{y_1, \dots, y_d\}$. We have $N(y_i) \cap X_1 = X_{b_i}$. So the bipartite graph induced by $X_1 \cup Y$ is isomorphic to G' . \square

Note that the presence of x_{d+1}, \dots, x_{2d} is useful if two vertices of B are twins with respect to A , meaning that their neighborhoods in A are the same, call it N . Then, even if X is shattered we are not sure that there exist two hyperedges intersecting A in exactly N (there is no notion of cardinality in the VC-dimension). Thus the vertices x_{d+1}, \dots, x_{2d} are there for ensures that all the vertices of Y have distinct neighborhoods on the set X . In fact, only $x_{d+1}, \dots, x_{d+\ln d}$ are needed to make $N(b_i)$ and $N(b_j)$ distinct: for $b_i \in B$, code i in binary over $\ln d$ bits and define $N(b_i)$ as to be the union of $\{x_{d+i_1}, \dots, x_{d+i_k}\}$ with the set of v_j such that the j -th bit is one. Thus the $2d$ upper bound of Lemma 3.9 can be improved into $d + \ln d$.

Lemma 3.9 can be reformulated as follows: if the VC-dimension of a hypergraph is bounded, then a “bipartite graph” is forbidden in the incidence bipartite graph. In particular, such a bipartite graphs appears with high probability as a subbipartite of a random graph that is large enough. In other words either the VC-dimension is arbitrarily large or large enough random-like bipartite graphs are forbidden. Since a random-like bipartite graph is forbidden for hypergraphs of bounded

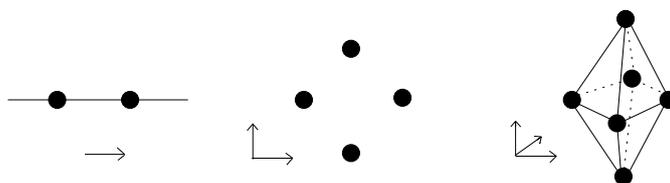


Figure 3.3: Sets of size $2d$ shattered by d -dimensional rectangles for $d = 1, 2, 3$.

VC-dimension, the structure of the incidence bipartite graph is constrained. The following sections explain how these constraints can be used in order to provide several properties of the hypergraph.

3.1.2 First examples

Geometrical hypergraphs. Intersections of geometrical objects are often “simple”, in the sense that they are constrained because of the topology and geometry of the space. Since the VC-dimension catches the complexity of the intersections of objects, it is natural to think that the VC-dimension must be bounded above in many cases.

Consider one of the easiest class of intersection of objects: the intersection of intervals in the real line (*i.e.* in \mathbb{R}^1). Vertices are points of the real line and intervals are represented by hyperedges. Let x_1, x_2, x_3 be three real numbers such that $x_1 \leq x_2 \leq x_3$. Every interval containing both x_1 and x_3 must contain x_2 (since an interval is convex). So $\{x_1, x_2, x_3\}$ cannot be shattered since no interval e satisfies $\{x_1, x_2, x_3\} \cap e = \{x_1, x_3\}$. Then the VC-dimension is at most 2.

More generally, the VC-dimension of intersection of axis-parallel d -dimensional rectangles is at most $2d$. A d -dimensional rectangle hypergraph is a hypergraph where the set of vertices V is a set of points of \mathbb{R}^d and a hyperedge corresponds to the intersection of an axis-parallel d -dimensional rectangle with V . Let X be a set of size $2d + 1$. Consider the set S containing all the vertices of X which are maximum or minimum for at least one of the d coordinates. If there are several minimum or maximum values, choose arbitrarily one of them. The set S contains at most $2d$ points (note that a same point can be maximum or minimum for several coordinates but it does not matter). Any rectangle containing all the points of S contains all the points of X which are between the first and the last point of each coordinate. In other words, any rectangle containing all the vertices of S also contains the whole set X . Hence X cannot be shattered since no rectangle has trace S .

Finally, one can verify that the VC-dimension is equal to $2d$ for some d -dimensional rectangle hypergraph. Indeed, if the set S defined in the previous paragraph has size $2d$, then X can be shattered. Figure 3.3 represents a set of size $2d$ which is shattered for $d = 1, 2, 3$. In order to shatter two new vertices in the new dimension, we just have to add one point above and below the previous structure (which are strictly between the first and the last values of the other coordinates).

Many other geometric classes have a bounded VC-dimension. For instance the VC-dimension of intersection of hyperplanes in d -dimensional space is $d + 1$, the VC-dimension of intersection of closed balls is $d + 2$. Convex sets with d faces in the plane have VC-dimension $2d$. Deeper examples of geometrical classes of bounded VC-dimension can be found in [124, 125, 160] for instance.

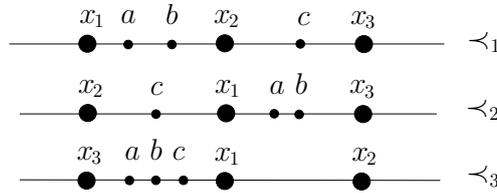


Figure 3.4: A 2-majority tournament. Vertices a and b are compatible for $\{x_1, x_2, x_3\}$. So they have the same in-neighborhood in $\{x_1, x_2, x_3\}$. Both a and b are not compatible with c for $\{x_1, x_2, x_3\}$.

Majority tournaments. Let $\prec_1, \dots, \prec_{2k-1}$ be total orders on V . The *majority tournament* on $\prec_1, \dots, \prec_{2k-1}$ is a tournament whose vertex set V such that xy is an arc if x is larger than y in at least k orders. A k -majority tournament T is a tournament such that there exist $2k-1$ orders $\prec_1, \dots, \prec_{2k-1}$ such that T is the majority tournament on $\prec_1, \dots, \prec_{2k-1}$. Note that 1-majority tournaments are transitive tournaments. Recall that, given a digraph $D = (V, A)$, the closed in-neighborhood hypergraph of D is the hypergraph whose vertex set is D and e is a hyperedge if $e = N^-[x]$ for some vertex x (recall that $N^-[x]$ is the closed in-neighborhood of x). The VC-dimension of a k -majority tournament is the VC-dimension of its closed in-neighborhood hypergraph.

k -majority
tournament

Lemma 3.10. *Every k -majority tournament has VC-dimension at most $(2k + o(1)) \log k$ where $o(1)$ tends to zero as k tends to infinity.*

Proof. Let $T = (V, E)$ be a k -majority tournament. Let X be a subset of vertices of size d . Two vertices y, z in V are incompatible for X if there exist an integer i and a vertex $x_j \in X$ such that $y \prec_i x_j \prec_i z$. Two vertices are compatible for X if they are not incompatible for X . The notions of compatibility and incompatibility are illustrated on Figure 3.4. Note that two compatible vertices have the same neighborhood in X . Indeed, for every vertex $x \in X$, two compatible vertices for X satisfy either $x \prec_i y, z$ or $y, z, \prec_i x$ for every i .

For every order \prec_i , the set X partitions the set $V \setminus X$ into $|X| + 1$ parts of consecutive vertices according to \prec_i . In some sense the vertices of the same class have the same behavior from the point of view of X in the order \prec_i . Note that two incompatible vertices are not in the same set of the partition for at least one order. Since there are $2k-1$ orders and since on each order partitions the vertex set into $d+1$ intervals, the number of classes is at most $(d+1)^{2k-1}$. Since two incompatible vertices are not in the same class, there are at most $(d+1)^{2k-1}$ pairwise disjoint incompatible vertices.

So if a X is shattered in the in-neighborhood hypergraph, it means that all the traces exist on X . Thus in particular we have $2^d \geq (d+1)^{2k-1}$ which provides the conclusion. \square

3.2 VC-dimension and number of hyperedges

This Section is devoted to proving that any hypergraph of bounded VC-dimension has a polynomial number of hyperedges. This result is known as Sauer's Lemma. Recall that general hypergraphs can have up to 2^n hyperedges. In order to prove it, we prove a stronger statement: the number of hyperedges can be bounded above by the number of shattered sets. We propose a classical proof of this result using a shifting argument. Section 3.2.2 provides a bijection between the edge set

and the set of ordered shattered sets. We will finally give a first application of the Sauer's lemma in Section 3.2.3.

3.2.1 The Sandwich Theorem and Sauer's Lemma

The Sandwich theorem bounds above and below the number of hyperedges in function of the number of shattered and strongly shattered sets. The proof presented here roughly follows the scheme of [99].

Theorem 3.11 (Sandwich Theorem). *Every hypergraph H satisfies*

$$ssh(H) \leq |E(H)| \leq sh(H).$$

Proof. Let us first prove that $|E(H)| \leq sh(H)$. The proof is based on a shifting technique. More precisely, the goal of the proof consists in constructing a hypergraph H^* such that:

- (i) H^* has as many hyperedges as H .
- (ii) If X is shattered in H^* , then X is shattered in H .
- (iii) The hyperedges of H^* are closed by inclusion. In other words, if e is a hyperedge of H^* then every subset of e is a hyperedge of H^* .

Note that such a hypergraph H^* satisfies $|E(H^*)| \leq sh(H^*)$. Indeed every hyperedge e is a shattered set in H^* since every subset of e is a hyperedge. The construction of H^* is based on a shifting technique described in Algorithm 3. One can easily verify that invariant (i) is satisfied by Algorithm 3 (no edge is collapsed into a smaller edge and no edge is deleted). The point (iii) is satisfied since it is the stop condition of Algorithm 3.

Algorithm 3: Shifting algorithm

Input : A hypergraph $H = (V, E)$.

Output: A hypergraph satisfying (i), (ii) and (iii) .

```

1 while The hypergraph is not closed by inclusion do
2   for  $(v \in V)$  do
3     for  $(e \in E)$  do
4       if  $(e \setminus \{v\} \notin E)$  then
5         Replace  $e$  by  $e \setminus \{v\}$ 

```

First note that Algorithm 3 ends since at each step, at least one hyperedge becomes strictly smaller. Let us prove that a shattered set after an application of line 5 is also shattered before this step. Let H be the hypergraph before the step of line 5 and H' be the hypergraph obtained after line 5 for the vertex v . Assume that a set X is shattered in H' . If $v \notin X$ then the set X is shattered in H since the hyperedges are only modified on v . So we can assume that $v \in X$. Let Y be a subset of X . If $v \in Y$ then the hyperedge with trace Y in H' has also trace Y in H (the hyperedge has not been modified). Assume now that $v \notin Y$. A hyperedge e has trace $Y \cup \{v\}$ in H' since X is shattered. Since the vertex $v \in e$ in H' , it means that e has not been modified during the step of the algorithm

between H and H' , *i.e.* there exists a hyperedge $e' = e \setminus v$ in H . Hence $e' \cap X = Y$. So X is also shattered in H , *i.e.* point (ii) holds.

The other inequality is obtained using the opposite graph. By Observation 3.6, a strongly shattered set is a set whose complement is not shattered in the opposite hypergraph (recall that the opposite hypergraph contains all the hyperedges which are not in H). In other words, $ssh(H) = 2^n - sh(H^c)$. And the number of edges of the opposite hypergraph is $2^n - |E|$. The inequality $|E(H^c)| \leq sh(H^c)$ ensures that $2^n - |E| \leq sh(H^c) = 2^n - ssh(H)$. Hence $ssh(H) \leq |E|$ which is the desired inequality. \square

Note that Theorem 3.11 still holds for hypergraphs induced by a subset of vertices since exactly the same method can be applied. Recall that a hyperedge e has trace X' on a set X if $e \cap X = X'$. Observation 3.1 provides an upper bound on the number of shattered sets when the VC-dimension is bounded above. By applying this inequality on Theorem 3.11 we obtain:

Lemma 3.12 (Sauer's Lemma). *Let $H = (V, E)$ be a hypergraph of VC-dimension d . For every set $X \subseteq V$, the number of (distinct) traces of E on X is at most $\sum_{i=0}^d \binom{|X|}{i}$. In particular, we have*

$$|E| \leq \sum_{i=0}^d \binom{n}{i}$$

Note that, in many cases, we do not need the exact upper bound on the number of hyperedges, but just an upper bound. In these cases, we will implicitly use the following inequality:

$$\sum_{i=0}^d \binom{|X|}{i} \leq k^{d+1}.$$

This lemma is known as Sauer's Lemma, or Sauer-Shelah's Lemma. It was proved independently by Sauer [178] and by Shelah [182]. A weaker version was also provided by Vapnik and Chervonenkis [190]. The Sauer's lemma proofs presented in these articles are purely combinatorial proofs and are based on induction. Other proofs of the same type can be found in [14, 26, 76, 108], some providing the reverse inequality of the Sandwich Theorem, others not. The proof of Theorem 3.11 we presented is due to Frankl [99]. Other proofs using shifting techniques are due to Frankl and Pach [100] and Alon [4]. Aharoni, Linial and Meshulam were the first to explicit a natural injection from the edge set into shattered sets in [168]. Note that an (implicit) injection also exists in the proof of Theorem 3.11. In Section 3.2.2, we will provide a bijection between the edge set and particular shattered sets called ordered-shattered sets. Such a notion of shattered set was studied in [14].

Let us compare Theorem 3.11 and Lemma 3.12. Even if Sauer's lemma will be enough for most of the following applications, the Sandwich Theorem can provide an arbitrarily better bound than Sauer's Lemma. For instance, Observation 3.2 ensures that some hypergraphs of VC-dimension $\log n$ have n shattered sets and n hyperedges. On such hypergraphs, the Sandwich Theorem is tight while the bound provided by Sauer's lemma is not even polynomial.

Nevertheless the gap between $sh(E)$ and $|E|$ can also be arbitrarily large. Let us slightly modify the example of Observation 3.2 in order to show it. Let V be a set of size n . Partition V into $\lfloor n/\log n \rfloor$ sets of size (at least) $\lfloor \log n \rfloor$. Since this example is quite "informal", we will forget the $\lfloor \rfloor$ in the following. Denote these sets by $(V_j)_{j \leq n/\log n}$ and by $v_1^j, \dots, v_{\log n}^j$ the vertices of V_j . For every

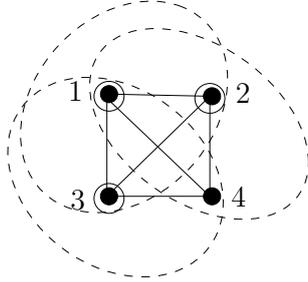


Figure 3.5: Hyperedges of size 2 are represented with lines. And hyperedges of size 3 with dashed hyperedges for readability.

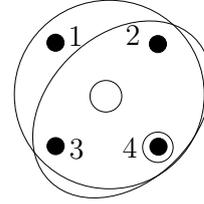


Figure 3.6: For readability we also represent the opposite hypergraph of Figure 3.5.

subset I' of $I = \{1, \dots, \log n\}$, construct the hyperedge containing the vertices v_i^j for all j and for all $i \in I'$. Note that the hypergraph contains exactly n hyperedges since there is exactly one hyperedge for each subset of I . In addition, one can easily check that every set $\{v_{i_1}^{j_1}, \dots, v_{i_\ell}^{j_\ell}\}$ is shattered as long as indices i_1, \dots, i_ℓ are pairwise distinct. Therefore the number of shattered sets is at least

$$\sum_{i=0}^{\log n} \binom{\log n}{i} (n/\log n)^i.$$

So the number of shattered set in not even polynomial in the number of hyperedges (which is equal to n).

Theorem 3.11 raises a notion of extremality. A hypergraph is *shattering extremal*, or *s-extremal* for short, if $|E| = sh(H)$. Shattering extremal hypergraphs have not been studied a lot (see [154] for instance). Nevertheless some nice questions have been raised, for instance:

Problem 5 (Mészáros and Rónyai). *Let H be a s-extremal hypergraph. Does there still exists in H a hyperedge e such that $H[E \setminus e]$ is s-extremal?*

Note that, since $sh(H[E \setminus e]) \leq sh(H)$ (a shattered set is still shattered in a larger hypergraph), we have $sh(H[E \setminus e])$ equals $|E \setminus e|$ or equals $|E \setminus e| + 1$ for every edge of a s-extremal hypergraph. Mészáros and Rónyai asked if there always exist an edge satisfying the equality.

The “dual question” admits a negative answer. More precisely, the dual question is the following: when $|E(H)| < sh(H)$, is it still possible to add a hyperedge to H in such a way the number of shattered sets does not increase? It would roughly mean that every hypergraph can be completed into a shattering-extremal hypergraph. The answer is negative, and Figure 3.5 gives a counter-example provided by Tom Drummond. In Figure 3.6, we also provide a figure of the opposite hypergraph which is more readable. One can verify that only $\{123, 234, 1234\}$ are not shattered (*i.e.* there are 13 shattered sets) and the hypergraph has 12 hyperedges. In addition, adding any missing hyperedge creates a new shattered set. Indeed the sets \emptyset and 4 shatter 123. The sets 234 and 1234 shatter 234.

In the following, we study a little bit later the structure of hypergraphs such that $|E(H)| = \sum_{i=0}^d \binom{n}{i}$.

Extremal hypergraphs. We just dealt with shattering extremal hypergraphs. This paragraph deals with a more classical notion of extremality. Let H_t be a hypergraph. In extremal theory, the goal is to determine the maximum number of hyperedges contained in a hypergraph with no copy of H_t . A hypergraph *contains a copy of H_t* if H_t can be obtained from H by a deletion of vertices and of hyperedges. Note that a hypergraph containing a copy of H_t is a hypergraph with a subset X of size $|H_t|$ such that the traces on X contains H_t (in terms of hyperedges). A hypergraph is *H_t -free* if it does not contain any copy of H_t . Let us denote by $ex(n, H_t)$ the maximum number of hyperedges of a hypergraph on n vertices which does not contain any copy of H_t . A hypergraph is *extremal (for H_t)* if it does not contain any copy of H_t and it has exactly $ex(n, H_t)$ hyperedges. Extremal hypergraphs have been extensively studied (see [71, 92, 133] for instance). A nice survey (though not recent) on extremal problems can be found in [101].

 H_t -free

Lemma 3.12 can be rephrased as follows: every hypergraph H with no copy of \mathcal{C}_{d+1} has at most $\sum_{i=0}^d \binom{n}{i}$ hyperedges. In addition, Lemma 3.12 is tight for several hypergraphs. Consider for instance the hypergraph H containing all the hyperedges of size at most d . By Observation 3.4, the VC-dimension of H is at most d . And the number of edges is exactly $\sum_{i=0}^d \binom{n}{i}$. Hence $ex(n, \mathcal{C}_{d+1}) = \sum_{i=0}^d \binom{n}{i}$.

It is straightforward that $ex(n, \mathcal{C}_{d+1}) \geq ex(n, \mathcal{U}_{\ell, d+1})$ for every ℓ . More precisely we have $ex(n, \mathcal{C}_{d+1}) \geq ex(n, F)$ for every F on $(d+1)$ vertices. Indeed if a hypergraph contains a shattered set X of size $d+1$, then all the traces of size ℓ exist on X , in particular those of F . The surprising point is that the reverse inequality is also correct in the uniform case. In other words, $ex(n, \mathcal{C}_{d+1}) = ex(n, \mathcal{U}_{\ell, d+1})$ for every $\ell > 0$.

Lemma 3.13. (Füredi, Quinn [102]) *For every ℓ, d , there exist $\mathcal{U}_{\ell, d+1}$ -free hypergraphs with exactly $\sum_{i=0}^d \binom{n}{i}$ hyperedges.*

Proof. Construct the following hypergraph H on the ordered vertex set x_1, \dots, x_n . First H contains all the hyperedges of size at most $\ell - 1$. In addition, H contains all the hyperedges e such that there exists an integer j satisfying $e \cap \{x_1, \dots, x_j\} = \ell - 1$ and e misses at most $d - \ell$ vertices in $\{x_{j+1}, \dots, x_n\}$.

Let X be a set of size $d+1$. Denote by $x_{i_1}, \dots, x_{i_{d+1}}$ the vertices of X in the increasing order. No hyperedge has trace $\{x_{i_1}, \dots, x_{i_\ell}\}$. Indeed, consider a hyperedge e containing $\{x_1, \dots, x_\ell\}$. By construction, there exists an integer j with $j \leq i_\ell$ such that all the vertices of index at least j except at most $d - \ell$ are in e . So at least one vertex of $x_{\ell+1}, \dots, x_{d+1}$ is in e .

Let us finally count the number of hyperedges. For every set x_{i_1}, \dots, x_{i_k} of size at most d , the set e such that $e \cap \{x_1, \dots, x_{i_\ell}\} = \{x_{i_1}, \dots, x_{i_\ell}\}$ and $e \cap \{x_{i_{\ell+1}}, \dots, x_n\} = \{x_{i_{\ell+1}}, \dots, x_n\} \setminus \{x_{i_{\ell+1}}, \dots, x_{i_k}\}$ is a hyperedge of H . If $d \leq \ell$ we assume that $x_{i_\ell} = x_n$ in the first equality. Note that all the hyperedges defined in this way are in the hypergraph and are distinct. So the hypergraph contains at least $\sum_{i=0}^d \binom{n}{i}$ hyperedges. And by Lemma 3.12, it contains exactly this number of hyperedges. \square

Problem 6. *Can we characterize the set of hypergraphs which are \mathcal{C}_d -extremal, or the set of hypergraphs which are \mathcal{U}_d -extremal?*

Problem 6 is a broadly open problem. Füredi and Quinn conjectured in [102] that all $\mathcal{U}_{\ell, d+1}$ -extremal hypergraphs have the same number of hyperedges of a given size. Several upper bounds on $ex(n, H)$ can be found in [13] for particular (small) hypergraphs H .

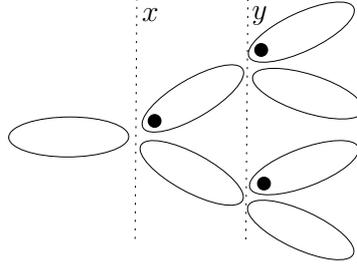


Figure 3.7: $\{x, y\}$ is an ordered shattered set, so the set $\{x, y\}$ is shattered.

3.2.2 Ordered shattered sets

In this section, we introduce and study ordered shattered sets. We will first see that ordered shattered sets are shattered. We will then prove that ordered shattered sets are in bijection with hyperedges.

Let $H = (V, E)$ be a hypergraph. Let $<$ be a total order on V . We denote by v_1, \dots, v_n the vertices of the graph such that $v_i < v_j$ if $i < j$. Two hyperedges e and e' are k -coherent if $e \cap \{v_1, \dots, v_k\} = e' \cap \{v_1, \dots, v_k\}$. In other words, we cannot distinguish e and e' if we just look at the k first vertices. Let $X = \{v_{i_1}, \dots, v_{i_\ell}\}$ be a subset of V . The set X is an *ordered shattered set* if there exists a set F of 2^ℓ hyperedges which can be refined in the following way: for every $j \leq \ell$, there exists a partition \mathcal{F}_j of F such that the partition \mathcal{F}_0 has exactly one set which is F and for every $0 \leq j \leq \ell$ the partition \mathcal{F}_j satisfies:

ordered
shattered set

- All the hyperedges in a same set of the partition of \mathcal{F}_j are $v_{i_{j+1}}$ -coherent.
- For every set of the partition \mathcal{F}_j , an half of the hyperedges contains v_{i_j} and the other does not contain it. It naturally refines each set of \mathcal{F}_j into two sets in \mathcal{F}_{j+1} .

The set F is called a *set associated to X* . Let us illustrate the notion of ordered shattered set. Let v_1, v_2, v_3 be three vertices and $e_1 = 101$ and $e_2 = 110$ be two hyperedges (where the i -th digit equals 1 if and only if v_i is in the hyperedge). The set $\{v_3\}$ is shattered since $e_1 \cap v_3 = v_3$ and $e_2 \cap v_3 = \emptyset$. But $\{v_3\}$ is not an ordered shattered set since e_1 and e_2 are not 2-coherent. Though $\{v_2\}$ is an ordered shattered set since e_1 and e_2 are 1-coherent (they both contain v_1) and $v_2 \in e_2$ and $v_2 \notin e_1$. The notion of refinement of the partition is illustrated on Figure 3.7.

Note that every set of the partition \mathcal{F}_j has size exactly $2^{\ell-j}$. Indeed the unique set of \mathcal{F}_0 has size 2^ℓ and at each step each set of the partition \mathcal{F}_{j-1} is divided into exactly two equal subsets. Hence after ℓ steps, all the sets have size exactly one.

Observation 3.14. *Every ordered shattered set is shattered.*

Proof. Let $X = \{v_{i_1}, \dots, v_{i_\ell}\}$ be an ordered shattered set and F be a set associated to X . Let \mathcal{F}_j be the partition after j steps. Let X' be a subset of X . Let us prove by induction on j that exactly one set of the partition \mathcal{F}_j satisfies $F \cap \{v_{i_1}, \dots, v_{i_j}\} = X' \cap \{v_{i_1}, \dots, v_{i_j}\}$. For $j = 0$, the result immediately holds. Let $j \leq \ell - 1$. By induction hypothesis there exists $F_j \subseteq \mathcal{F}_j$ satisfying the conditions. By assumption, F_j is divided into two equal parts in \mathcal{F}_{j+1} , one containing $v_{i_{j+1}}$ and not the other one. So depending if X' contains or not the vertex $v_{i_{j+1}}$, we choose one or the other subset of \mathcal{F}_{j+1} which refines F_j .

Since all the sets of the partition \mathcal{F}_ℓ have size one, the induction ensures that a hyperedge e satisfies $e \cap X = X'$. So the set X is shattered. \square

Before proving that there exists a bijection between ordered shattered set and hyperedges, let us introduce a new notion. A hyperedge e *collapses on* v_k if there exists a hyperedge e' such that e and e' are $(k-1)$ -coherent, and $v_k \in e$ and $v_k \notin e'$. Roughly speaking, a hyperedge e collapses on v_k if there exists a hyperedge e' such that e and e' are distinct for the first time on v_k . The *collapse index* of e is the maximum k such that e collapses on v_k . The *collapsed hypergraph on* v_k , denoted by $C(v_k)$ is the hypergraph on vertex set $\{v_1, \dots, v_{k-1}\}$ whose hyperedges are restrictions on $\{v_1, \dots, v_{k-1}\}$ of the hyperedges collapsing on v_k . The *collapsed hypergraph on* $v_{i_1}, \dots, v_{i_\ell}$ (where $v_{i_1} < \dots < v_{i_\ell}$) denoted by $C(v_{i_1}, \dots, v_{i_\ell})$ is the collapsed hypergraph on v_{i_1} of the collapsed hypergraph on $v_{i_2}, \dots, v_{i_\ell}$.

Let $i < j$. If a hyperedge e collapses in v_i and v_j , then e does not necessarily collapse on v_i in the collapse hypergraph on v_j . Indeed consider the three hyperedges $e_1 = 11$, $e_2 = 10$ and $e_3 = 01$. One can note that e_1 collapses with e_3 on v_1 and with e_2 on v_2 . Nevertheless, the collapse hypergraph on v_2 only contains the hyperedge e_1 (since e_3 does not collapse on v_2). So e_1 does not collapse on the collapse hypergraph on v_2 .

Let e be a hyperedge. Let us define inductively the *collapse sequence of* e . If e never collapses, the collapse sequence is the empty sequence. Otherwise denote by k the collapse index. The collapse sequence of e is the collapse sequence of e in $C(v_k)$ concatenated with the collapse index k . Informally, the collapse sequence contains the collapse index and then we repeat the operation by induction on the collapse hypergraph on v_k . Note that by construction all the vertices of the collapse sequence of e are in the hyperedge e . Let us first prove that distinct hyperedges have distinct collapse sequences.

Lemma 3.15. *Two distinct hyperedges have distinct collapse sequences.*

Proof. Assume by contradiction that two distinct hyperedges e and e' have the same collapse sequence $v_{i_1}, \dots, v_{i_\ell}$. Let us denote by v_k the first vertex in the order $<$ such that e and e' are distinct (so $(k-1)$ is the maximum integer such that e and e' are $(k-1)$ -coherent). Free to exchange e and e' , we may assume that $v_k \in e$ and $v_k \notin e'$. Let j be the integer such that $i_{j-1} \leq k \leq i_j$ where i_0 denotes 0 and $i_{\ell+1}$ denotes $+\infty$.

Note that both e and e' are in the collapse hypergraph $C(v_{i_j}, \dots, v_{i_\ell})$ since e and e' have collapse sequence $v_{i_1}, \dots, v_{i_\ell}$. Since $v_{i_{j-1}}$ is in the collapse sequence of both e and e' , the vertex $v_{i_{j-1}}$ is in both e and e' . So k is strictly larger than i_{j-1} . Since e and e' are $(k-1)$ -coherent and are distinct on v_k , the hyperedge e collapses on v_k on $C(v_{i_j}, \dots, v_{i_\ell})$, a contradiction with the maximality of i_{j-1} . \square

Since a collapse sequence is associated to a hyperedge, Lemma 3.15 ensures that there exists a bijection between collapse sequences and hyperedges. Let us finally prove that there is a bijection between collapse sequences and ordered shattered sets.

Lemma 3.16. *Every collapse sequence is an ordered shattered set.*

Proof. Assume that a hyperedge e has collapse sequence v_{i_1}, \dots, v_{i_k} . Let us prove that $X = \{v_{i_1}, \dots, v_{i_k}\}$ is an ordered shattered set. More precisely let us prove by induction that for every j , there are 2^j hyperedges F_j which are associated to $\{v_{i_1}, \dots, v_{i_j}\}$ on the collapse hypergraph

$C(v_{i_{j+1}}, \dots, v_{i_k})$. For $j = 0$, any hyperedge e in $C(v_{i_1}, \dots, v_{i_k})$ works since $e \cap \emptyset = \emptyset$. Such a hyperedge exists since there exists a hyperedge e with collapse sequence v_{i_1}, \dots, v_{i_k} (and then $C(v_{i_1}, \dots, v_{i_k})$ is not empty). So we put $F_0 = \{e\}$.

Let $j \leq k - 1$. Denote by F_j the set of hyperedges associated to $\{v_{i_1}, \dots, v_{i_j}\}$ on the collapse hypergraph $C(v_{i_{j+1}}, \dots, v_{i_k})$. And denote by \mathcal{F}_ℓ^j its refinement partition for $\ell \leq j$. Let F_{j+1} be a set of hyperedges such that each hyperedge e of F_j is extended into two hyperedges in $C(v_{i_{j+2}}, \dots, v_{i_k})$, one containing $v_{i_{j+1}}$ and not the other one. The set F_{j+1} has size 2^{j+1} . For every set $F \in \mathcal{F}_\ell^j$ with $\ell \leq j$, we create the set $F' \in \mathcal{F}_\ell^{j+1}$ such that F' contains both extensions of all the hyperedges e of F . Thus, by induction, any set of \mathcal{F}_ℓ^{j+1} has size $2^{j+1-\ell}$ and the hyperedges are still $v_{i_{\ell-1}}$ -coherent. And the partition \mathcal{F}_{j+1}^{j+1} is the partition with exactly one hyperedge of F_{j+1} in each set. After k steps, the set F_k of hyperedges is a set associated to v_{i_1}, \dots, v_{i_k} on the whole hypergraph, and then v_{i_1}, \dots, v_{i_k} is an ordered shattered set. \square

Let us now prove the converse of Lemma 3.16.

Lemma 3.17. *Every ordered shattered set is a collapse sequence.*

Sketch of the proof. Let us prove by induction on decreasing j that if $X = \{v_{i_1}, \dots, v_{i_k}\}$ is an ordered shattered set then v_{i_1}, \dots, v_{i_j} is an ordered shattered set on $C(v_{i_{j+1}}, \dots, v_{i_k})$. Let X be an ordered shattered set. If $j = k$ then the conclusion holds.

Let F be a set of hyperedges associated to $v_{i_1}, \dots, v_{i_{j+1}}$ on $C(v_{i_{j+2}}, \dots, v_{i_k})$. Our proof is based on the switching algorithm in order to find the good set F associated to v_{i_1}, \dots, v_{i_j} . Let e be a hyperedge of F containing v_{i_k} . Let us prove that we can assume that the collapse index of e is i_j and then that e is in the collapse hypergraph on v_{i_j} . Choose e in such a way the first collapse of e after v_{i_j} is maximized (we assume that it is infinite if such a collapse does not exist). If it is infinite, the conclusion holds. Indeed i_j is the collapse index of e and then e is in the collapse hypergraph on v_{i_j} .

Assume by contradiction that e collapse on v_ℓ . So there exists a hyperedge e' on $C(v_{i_{j+2}}, \dots, v_{i_k})$ such that e and e' are $(\ell - 1)$ -coherent and e' does not contain v_ℓ . Since e does not collapse between v_{i_j} and $v_{\ell-1}$ the same holds for e' and e' does not collapse on v_ℓ since $v_\ell \notin e'$. Finally the first collapse of e' after v_{i_j} is strictly after the first collapse of e , a contradiction with the maximality of e . So, free to exchange hyperedges of F , we can assume that hyperedges of F do not collapse after v_{i_j} and then all the hyperedges of F containing v_{i_j} are on the collapse hypergraph on v_{i_j} . So $v_{i_1}, \dots, v_{i_{j-1}}$ is an ordered shattered set in $C(v_{i_j})$.

Let us finally prove that every ordered shattered set is a collapse sequence. If the sequence is empty the conclusions holds. Otherwise, let $X = \{v_{i_1}, \dots, v_{i_k}\}$. We have just seen that $X \setminus v_{i_k}$ is an ordered shattered set on $C(v_{i_k})$. So by induction it is a collapse sequence on $C(v_{i_k})$. Extend e on the whole vertex set in such a way e contains v_{i_k} (which is possible by using the same argument as in the first part of the proof). Then the collapse sequence of e is X . \square

Combining Lemmas 3.15, 3.16 and 3.17, we obtain the following theorem which strengthens Lemma 3.12.

Corollary 3.18 (Anstee et al. [14]). *Let H be a hypergraph. There exists a bijection between the hyperedges of H and the ordered shattered sets of H .*

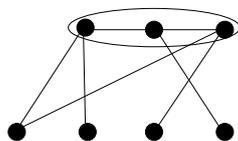


Figure 3.8: The three vertices at the top form an identifying code of the whole graph on 7 vertices.

In addition, the collapse sequence of a hyperedge can be found in polynomial time. So this bijection can be found in polynomial time.

Up to my knowledge, no proof using VC-dimension is based on this natural bijection between the set of hyperedges and ordered shattered sets. This bijection could be interesting in particular when a natural order is inherited from the vertex set of the input structure. In Section 3.2.3, we give an application of Lemma 3.12 which is a weaker version of Corollary 3.18

3.2.3 Application to identifying codes

Let us now provide a short application of Lemma 3.12 to identifying codes. Let G be a graph. A subset X of vertices is an *identifying code* if for every pair of vertices $v, v' \in V$, we have $N[v] \cap X \neq N[v'] \cap X$ and in addition the set X is a dominating set. Recall that $N[v]$ denotes the closed neighborhood of v . Two vertices x, y are true twins if $N[x] = N[y]$. A graph containing true twins does not admit any identifying code. Conversely, if G does not contain true twins, one can easily verify that the whole vertex set is an identifying code. Given a set X of vertices, u and v are *identified by X* if $N[u] \cap X \neq N[v] \cap X$. In other words, all the hyperedges of the closed neighborhood hypergraph have a distinct trace on the identifying code. In Figure 3.8, the above set is an identifying code of size 3. Note that the size of the identifying code is optimal since an identifying code has size at least $\lceil \log n - 1 \rceil$. Indeed every vertex must have a distinct neighborhood on the identifying code and there are $2^{|X|}$ possible neighborhoods on a set of size X (and the minus one came from the fact that the empty neighborhood is not authorized). Note also that vertices in the identifying code also have to be identified by the identifying code.

Introduced in 1998 in [126], identifying codes received since an accurate attention (see [17, 47, 95] for instance). Most of the known results hold for particular classes of graphs. The following result due to Aurélie Lagoutte, Zhentao Li, Aline Parreau and Stéphan Thomassé and myself ensures that classes of graphs closed by induced subgraphs satisfy a dichotomy-like theorem: either arbitrarily large graphs admit logarithmic identifying codes or all the graphs of the class admit polynomial (*i.e.* of size n^ϵ for some $\epsilon > 0$) identifying codes.

Theorem 3.19 (B., Lagoutte, Li, Parreau, Thomassé). *Every class of graph \mathcal{C} closed by induced subgraphs satisfies:*

1. *Either for every k , there exists a graph $G_k \in \mathcal{C}$ with more than $2^k - 1$ vertices with an identifying code of size $2k$.*
2. *Or there exists $\epsilon > 0$ such that for every twin-free graph G of \mathcal{C} , the minimum size of an identifying code is at least $|V|^\epsilon$.*

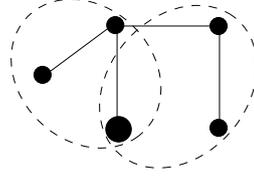


Figure 3.9: The big vertex is a $1/2$ -net (with the uniform measure) but not a hitting set

Proof. Let \mathcal{C} be a class of graphs closed by induced subgraphs. Assume first that for any $k \in \mathbb{N}$, there is a graph of \mathcal{C} with VC-dimension at least k (\mathcal{C} has infinite VC-dimension). Then we show that \mathcal{C} satisfies case 1.

Indeed, let $k \in \mathbb{N}$. Let H be a graph of \mathcal{C} with VC-dimension k . Let X be a shattered set of size k . Let Y be a set of $2^k - 1$ vertices of H that shatters all the non-empty subsets of X . Choose Y in such a way that $|X \cap Y|$ is maximized. Then pairs of vertices between X and $Y \setminus X$ have distinct neighborhoods on X (otherwise a vertex of $Y \setminus X$ can be replaced by a vertex of X and then $|Y \setminus X|$ decreases). Let $G_k = H[X \cup Y]$. Note that $G_k \in \mathcal{C}$ since \mathcal{C} is closed by induced subgraphs. The graph G_k has at least $2^k - 1$ vertices since the set Y contains at least $2^k - 1$ vertices. We have seen that any pair of vertices in X and $Y \setminus X$ are identified by X . In addition any pair y, y' in Y is identified by X since by definition of Y , we have $N_{G_k}[y] \cap X \neq N_{G_k}[y'] \cap X$ if $y \neq y'$. Let u be a vertex of X and let $y_u \in Y$ such that $N[y_u] \cap X = \{u\}$ (such a vertex exists by definition). Let Z be the union of the vertices y_v for every $v \in X$. Let u, v be two distinct vertices of X . The vertices u and v are identified by Z since for every $u \in X$, we have $N[u] \cap Z = \{y_u\}$. So the set $Z \cup X$ is an identifying code of G_k of size at most $2k$.

Assume now that every graph in \mathcal{C} has VC-dimension at most d . Let G be a twin-free graph of \mathcal{C} on n vertices. Let C be an identifying code of G . All the traces of vertices of G on C must be different. Hence, Lemma 3.12 ensures that $n \leq \sum_{k=0}^d \binom{|C|}{k} \leq |C|^{d+1}$. Therefore, $|C| \geq n^{1/(d+1)}$, proving that \mathcal{C} satisfies the second claim. \square

3.3 VC-dimension and integrality gap

In Chapter 2.2, we have seen that the gap between the transversality τ and the fractional transversality τ^* can be arbitrarily large. For instance, the complete hypergraph $\mathcal{U}_{n+1, 2n}$ satisfies $\tau = n$ and $\tau^* \leq 2$ by Lemma 2.18. This hypergraph has VC-dimension $n - 1$ by Observation 3.5. In this part, we will see that it is not surprising in the following sense: if the integrality gap between τ and τ^* is arbitrarily large, then the VC-dimension must be large.

In Section 3.3.1, we state the main result of VC-dimension which ensures that the size of ϵ -nets is bounded as long as the VC-dimension is bounded. We prove that it implies that the integrality gap is bounded. Section 3.3.2 is devoted to the first applications and remarks. Another application will be provided in Chapter 5.

3.3.1 Upper bounds on the size of ϵ -nets

A *measure* of a hypergraph is a weight (*i.e.* non negative) function on the vertex set such that the sum of the weights equals one. Let H be a hypergraph and μ be a measure on the vertex set of H . An ϵ -net is a subset of vertices X such that every hyperedge of weight at least ϵ intersects X . The *uniform measure* is the measure where all the vertices are given the same weight (*i.e.* weight $1/|V|$). In this case, an ϵ -net is a subset of vertices intersecting every hyperedge of size at least ϵn . In Figure 3.9, the big vertex is an $1/2$ -net (for the uniform measure) since it intersects all the hyperedges of size at least $1/2$. Note that it is not a hitting set. On the contrary, a $2/5$ -net would be a hitting set of the hypergraph. More generally, let c be the minimum size of a hyperedge, a c/n -net is a hitting set (for the uniform measure). A key result of Haussler and Welzl [117] in VC-dimension theory bounds above the minimum size of an ϵ -net in function of the VC-dimension and of ϵ .

Theorem 3.20. [Haussler, Welzl [117]] *Every hypergraph of VC-dimension d has an ϵ -net of size $\mathcal{O}(\frac{d \ln(d/\epsilon)}{\epsilon})$.*

Sketch of the proof. The proof of Theorem 3.20 is tricky. In the following the main steps of the proof are presented, but for simplicity, technical details and calculations are omitted. For simplicity, we assume that the measure is uniform. The proof for non-uniform measures follows the same scheme but is a slightly more technical. For a whole proof the reader is referred to [71, 117, 150] for instance. We can assume without loss of generality that all the hyperedges have size at least ϵn . Indeed the others do not have to intersect the ϵ -net so we omit them.

Draw a random vertex subset X of size $s := C \cdot (d/\epsilon) \ln(d/\epsilon)$ where C is a positive constant not detailed here. The proof is devoted to showing that X is an ϵ -net with positive probability. For every hyperedge e and every vertex $x \in X$, we have $\mathbb{P}(x \in e) \geq \epsilon$ since e contains at least ϵn vertices and X is drawn randomly. Since vertices of X are picked independently, the average number of vertices of X in e is at least $s \cdot \epsilon$. So, for every $e \in E$, Tchebychev inequality ensures that $\mathbb{P}(|e \cap X| \geq s\epsilon/2) \geq 1/2$. A hyperedge e *heavily intersects* X if $|e \cap X| \geq s\epsilon/2$.

Let us call E_0 the event “there exists a hyperedge which is not intersected by X ”. Note that E_0 is the complement event of our objective. Since we want to show that the complement of E_0 has a strictly positive probability, let us prove that $\mathbb{P}(E_0) < 1$. Draw randomly a second vertex subset Y of size s . Call E_1 the event “there exists a hyperedge which is not intersected by X and which heavily intersects Y ”. We have $\mathbb{P}(E_1) \leq \mathbb{P}(E_0)$ since E_1 requires E_0 . And we also have $\mathbb{P}(E_0) \leq 1/2 \mathbb{P}(E_1)$. Indeed, informally, if X is not an ϵ -net, then a hyperedge e is not intersected by X and e has probability at least $1/2$ to be heavily intersected by Y (since X and Y are drawn independently).

The rest part of the proof consists in proving that $\mathbb{P}(E_1) < 1/2$, which implies that $\mathbb{P}(E_0) < 1$. It finally ensures that there is a positive probability that X is an ϵ -net, *i.e.* at least one set of size s is an ϵ -net.

Let $k = s\epsilon/2$. Let A be a set of size $2s$. Keep in mind that the first half of A corresponds to X and the second half corresponds to B . Let us now draw a subset Z of size k in A . We want to compute the probability P that no vertex of Z appears in the first part of A (*i.e.* the probability that a hyperedge does not intersect X but heavily intersects Y). It is equivalent to force a set Z of size k to be in the second part of an equal bipartition of A . There are $\binom{2s-k}{s}$ bipartitions in which Z is in the second

ϵ -net
uniform
measure

Haussler
Welzl

part. And the total number of bipartitions equals $\binom{2s}{s}$. So:

$$P = \frac{\binom{2s-k}{s}}{\binom{2s}{s}} \leq (1/\epsilon)^{-Cd/4}.$$

The last inequality is not immediate but can be obtained by a non trivial sequence of calculations (no deep argument is used at this point). Note that, until now, no argument based on VC-dimension has been used. Let us now link the events E_0 and E_1 . Without the VC-dimension, the unique thing we can claim is that, since there are m hyperedges, $\mathbb{P}(E_0) \leq (1/\epsilon)^{-Cd/4} m$. But since the VC-dimension is at most d , Lemma 3.12 ensures that the number of traces on A is at most $|A|^{d+1}$. In other words, we have $\mathbb{P}(E_0) \leq (1/\epsilon)^{-Cd/4} \cdot (2s)^{d+1}$. A last sequence of non-trivial calculations ensures that $\mathbb{P}(E_1) < 1/2$. So $\mathbb{P}(E_0) < 1$, which achieves the proof. \square

The proof ensures that when we pick randomly vertices, then the probability of finding an ϵ -net is positive. Hence it provides an randomized approximation algorithm for finding transversal in a hypergraph of bounded VC-dimension. Deterministic proofs (and algorithms) of Theorem 3.20 also exist (a derandomized proof is proposed in [149]).

Note that the size of an ϵ -net can be at least $1/\epsilon$. Indeed, the size of an ϵ -net of a hypergraph containing $1/\epsilon$ disjoint hyperedges of size ϵn is at least $1/\epsilon$ (an ϵ -net must intersect each of the $1/\epsilon$ disjoint hyperedges). In [117], Haussler and Welzl asked whether the logarithmic factor can be removed from Theorem 3.20. Pach and Woeginger proved in [162] that this factor can be eliminated for hypergraphs of VC-dimension one. In the other cases, Komlós et al. [128] and Pach and Agarwal [160] closed the gap between lower and upper bounds. More precisely we have the following:

Theorem 3.21. *Let $d \geq 2$. Denote by $s(d, n)$ the maximum size of a minimum ϵ -net over the set of hypergraphs of VC-dimension d on n vertices. We have:*

$$\frac{(d-2 + \frac{1}{d+2} + o(1)) \ln(1/\epsilon)}{\epsilon} \leq s(d, n) \leq \frac{(d + o(1)) \ln(1/\epsilon)}{\epsilon}.$$

where $o(1)$ is a function which tends to zero when n tends to infinity.

Theorem 3.20 can be rephrased in order to obtain a bounded gap between τ and τ^* . Recall that τ denotes the minimum size of a hitting set and τ^* its fractional relaxation (see Section 2.2). In the following we will exclusively use this formulation instead of the (although stronger) formulation with ϵ -nets.

Corollary 3.22. *Every hypergraph of VC-dimension d satisfies:*

$$\tau \leq \mathcal{O}(d\tau^* \ln(d\tau^*)).$$

Proof. Let w be a weight function on the vertex set corresponding to an optimal solution of the transversal linear program, i.e. $\sum_{x \in V} w(x) = \tau^*$ and for every hyperedge e , $\sum_{v \in e} w(v) \geq 1$. Let μ be the weight function such that each vertex x satisfies $\mu(x) = w(x)/\tau^*$. Note that $\sum_{x \in V} \mu(x) = 1$, i.e. μ is a measure. Every hyperedge has weight at least $1/\tau^*$ for μ since every hyperedge has weight at least one for w . So a $(1/\tau^*)$ -net is a subset of vertices which intersects every hyperedge, i.e. a hitting set. Theorem 3.20 ensures that there is a hitting set of size $\mathcal{O}(d\tau^* \ln(d\tau^*))$. \square

Corollary 3.22 ensures that the integrality gap between τ and τ^* is bounded if the VC-dimension is bounded. Previous statements do not explicit constants. The following theorem, whose proof can be found in [71], explicit them.

Theorem 3.23. *Every hypergraph H with $|E(H)| \geq 2$ of VC-dimension d satisfies*

$$\tau \leq 2d\tau^* \log(11\tau^*).$$

Erdős-Pósa property. Since the integrality gap between τ and τ^* is bounded, Theorem 3.23 raises the following question: does the same hold for ν and ν^* ? Recall that ν denotes the packing number and ν^* its fractional relaxation. It would imply immediately the Erdős-Pósa property since $\tau^* = \nu^*$ by Theorem 2.16. Unfortunately, the following example ensures that the integrality gap for ν is not bounded, even for hypergraphs of VC-dimension 2.

Lemma 3.24. *There exist hypergraphs of VC-dimension 2 such that $\nu = 1$ and $\nu^* = \mathcal{O}(\sqrt{n})$.*

Proof. The counter-example is the same as in Lemma 2.19 which provides an arbitrarily large gap between ν and ν^* . Let K_n be a clique on n vertices. Construct a hypergraph H_n such that vertices of H_n correspond to edges of K_n . Hyperedges of H_n correspond to the subsets of edges adjacent to a same vertex. We have seen in Lemma 2.19 that $\nu(H_n) = 1$ and $\nu^*(H_n) = \mathcal{O}(n)$. In addition, every vertex of H_n is contained in exactly two hyperedges, so no set of size 3 is shattered. Indeed, given a set X of size k , exactly 2^{k-1} subsets of X contain a fixed element x . So if a set of size 3 is shattered, at least 4 hyperedges contain each of the shattered vertices, a contradiction. \square

3.3.2 Applications

In this Section, we give applications of Theorem 3.21 for problems on graphs. When the proofs are not too complicated we also give the proofs, or at least a sketch of the proofs of these results.

Geometrical hypergraphs. Theorem 3.21 ensures that there exist ϵ -nets of size $\mathcal{O}(1/\epsilon \log(1/\epsilon))$. Nevertheless, in geometrical cases, the upper bound on the minimum size of ϵ -nets can often be improved. For instance, linear upper bounds (*i.e.* bounds of size $\mathcal{O}(1/\epsilon)$) are known for intersection of halfspaces in dimension two and three, intersection of disks or pseudo-disks (see [61, 152, 167] for instance).

The existence of a better upper bound in the case of intersections of axis-parallel rectangles was a challenging problem for years. Aronov, Ezra and Sharir [15] recently proved that the upper bound can be improved into a $\mathcal{O}(1/\epsilon \log(\log(1/\epsilon)))$ upper bound (and the same holds for 3-dimensional rectangles). Their proof is based on [49] and on [61]. Pach and Tardos proved that such a bound is sharp [161].

On the contrary, the existence of geometrical examples for which the $\Omega(1/\epsilon \log(1/\epsilon))$ is reached was open for years. Pach and Tardos in [161] also provide such a geometrical example (of VC-dimension 2).

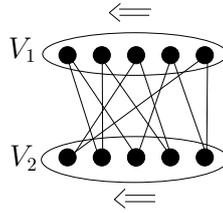


Figure 3.10: A tournament which is union of 3 partial orders with an arbitrarily large VC-dimension. The first order is a union of two total orders on V_1 and V_2 . The second is a random oriented bipartite graph from V_1 to V_2 . The last one is the remaining arcs from V_2 to V_1

k -majority tournaments. In this part, we study dominating sets on k -majority tournaments. Dominating sets for graphs were defined in Chapter 1, let us define it for digraphs. A dominating set X is a subset of vertices such that for any vertex $y \in V$, there exists a vertex $x \in X$ such that $y \in N^+[x]$. In other words, y is either in X or is a neighbor of a vertex of X . General tournaments can have dominating sets of arbitrarily large size, as first proved by Erdős in [86]. Kierstead and Trotter conjectured that every k -majority tournament has a dominating set of size $f(k)$. Alon and al. [6] answer positively to this question using Theorem 3.23.

Theorem 3.25 (Alon et al. [6]). *Every k -majority tournament has a minimum dominating set of size $\mathcal{O}(k \log k)$.*

Proof. The proof is based on several results we already mentioned. Let T be a k -majority tournament. Consider the closed in-neighborhood hypergraph of T . First note that a dominating set of T is a hitting set of H . Indeed consider a hitting set X of H . For every vertex y not in X , there exists a vertex $x \in X$ such that x is in the out-neighborhood of y , i.e. xy is an arc. Indeed y is in the in-neighborhood of x so x is in the out-neighborhood of y . So X is a dominating set of T . As H is the closed in-neighborhood of a k -majority tournament, Lemma 3.10 ensures that $vc(H) \leq \mathcal{O}(k \log k)$. And Lemma 2.23 ensures that $\tau^* \leq 2$. Indeed the weight function of Lemma 2.23 has total weight 2 and satisfies $N^+[x] \geq 1$ for every vertex (i.e. the constraints are satisfied). An application of Theorem 3.23 finally provides the conclusion of Theorem 3.25. \square

Lots of classes of tournaments are defined via orders. Two partial orders are *disjoint* if every pair x, y is comparable (i.e. $x < y$ or $y < x$) in at most one order. A tournament T is a disjoint union of k partial orders if there exist k disjoint partial orders $<_1, \dots, <_k$ such that, for every arc xy , exactly one order satisfies $y < x$. In other words, a tournament is a disjoint union of k partial orders if there exists a k -coloring of the tournament such that each color is an acyclic arc-transitive digraph (i.e. if xy and yz are arcs then xz is an arc).

Conjecture 4 (Gyárfás). *Every tournament which is a disjoint union of k partial orders has dominating set of size $f(k)$.*

This conjecture was studied in [163] by Pálvölgyi and Gyárfás. They proved that Conjecture 4 implies a generalization of the result of Bárány and Lehel on covering sets by boxes. In addition, Conjecture 4 is linked with the following conjecture.

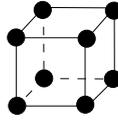


Figure 3.11: A cube

Conjecture 5 (Erdős, Sands, Sauer, Woodrow). *There exists a function f such that any k -arc-coloring of any tournament admits $f(k)$ vertices X such that every vertex can be reached with an edge-monochromatic path starting on X .*

Note that Conjecture 4 is a particular case of Conjecture 5. Indeed consider tournament which can be colored with k colors such that each color is a transitive digraph. Let X be a set of vertices such that every vertex can be reached with a monochromatic path starting on X . Then X is a dominating set of the tournament. Indeed let y be a vertex of the graph. There is an arc-monochromatic path from a vertex x of X to y . But since each color induces a transitive directed graph, there is an arc from x to y . Conjecture 5 received lots of attention (see [153, 177] for instance). Conjecture 4 holds for $k = 1, 2$ since such tournaments are transitive tournaments (and then $\tau = 1$). The conjecture is still open for $k \geq 3$.

As underlined in [163], the scheme of the proof of Theorem 3.25 cannot hold for proving Conjecture 4 since the VC-dimension of such hypergraphs can be arbitrarily large as long as $k \geq 3$. Consider the example of Figure 3.10. First partition the vertex set into two equal parts, V_1 and V_2 . The first order $<_1$ is a total order on V_1 union a total order on V_2 . The order $<_2$ is a random (oriented) bipartite graph from V_1 to V_2 . There is no problem of transitivity since there is no path of length 2. Arcs of $<_2$ are represented by edges in Figure 3.10. Non comparable vertices in $<_1$ and $<_2$ are pairs v_2, v_1 where $v_2 \in V_2$ and $v_1 \in V_1$. So we put $<_3$ as the “complement” bipartite graph of $<_2$ with arcs from V_2 to V_1 (arcs of $<_3$ are represented by non edges in Figure 3.10). Since there is no constraints on the bipartite graph from V_1 to V_2 , the VC-dimension of the in-neighborhood hypergraph can be arbitrarily large as underlined in Lemma 3.9.

Chromatic number of triangle-free graphs. In this part, we deal with a problem raised by Erdős and Simonovits concerning triangle-free graphs of large minimum degree. Erdős and Simonovits conjectured in [84] that every triangle-free graph of minimum degree at least $n/3$ has a bounded chromatic number. In addition, they showed that there exist graphs with minimum degree at least $(\frac{1}{3} - \epsilon)n$ with arbitrarily large chromatic number. Thomassen proved in [184] that every graph with minimum degree at least $(\frac{1}{3} + \epsilon)n$ have a bounded chromatic number. Finally Brandt and Thomassé [42] proved that all such graphs have chromatic number at most 4. Using VC-dimension, Luczak and Thomassé [140] finally broke the $\frac{n}{3}$ barrier (with an additive constant). The key lemma of their proof is the following. A *cube* is the graph represented in Figure 3.11.

Lemma 3.26 (Luczak, Thomassé [140]). *Every triangle-free graph G with minimum degree cn and no induced cube satisfies:*

$$\chi(G) \leq \frac{24 \cdot \log(11/c)}{c}.$$

Proof. Consider a maximum cut (X, Y) of the vertex set, *i.e.* a bipartition of the vertex set such that the number of edges between X and Y is maximized. Note that any vertex x of X has at least as many neighbors in Y as in X . Otherwise, we can put x in Y instead of X and (strictly) increase the size of the cut, a contradiction with the maximality of the cut. So every vertex of X has at least $cn/2$ neighbors in Y (since the minimum degree is at least cn). Consider the hypergraph H on vertex set Y where hyperedges are traces of neighborhoods of vertices of X in Y . In other words, Y' is a hyperedge if there exists a vertex x of X such that $N(x) \cap Y = Y'$.

Observation 3.27. *We have $\chi(G[X]) \leq \tau(H)$.*

Proof. Consider a hitting set Z of H . So, by definition of H , for every vertex $x \in X$, there exists $z \in Z$ such that $z \in N(x)$. It means that every vertex of X is in the neighborhood of a vertex of Z . So X can be covered by the neighborhoods of the vertices of Z . Since G is triangle-free each neighborhood is a stable set (otherwise there would be a triangle). Hence $\chi(G[X]) \leq |Z|$. \square

Let us finally prove that $\tau(H)$ is bounded. First note that by Observation 2.17 τ^* is at most $2/c$ as every hyperedge has size $cn/2$. Let us now show that the VC-dimension of H is at most 3. Assume by contradiction that the VC-dimension is at least 4. Denote by $Y' = \{y_1, y_2, y_3, y_4\}$ a shattered set of size four. In particular, there exist four vertices x_1, x_2, x_3, x_4 in X such that $N(x_i) \cap Y' = Y' \setminus \{y_i\}$. As G does not contain any triangle, the subgraph induced by these eight vertices is a cube, a contraction.

Since $\tau^*(H) \leq 2/c$ and $vc(H) \leq 3$, Theorem 3.23 ensures that H has a hitting set of size $12 \log(1/c)/c$. Let Z be such a hitting set. Observation 3.27 implies that $\chi(G[X]) \leq 12 \log(1/c)/c$. By symmetry the same holds for Y . And one can easily verify that if V_1, V_2 is a partition of the vertex set V then $\chi(G[V]) \leq \chi(G[V_1]) + \chi(G[V_2])$ (we just have to use distinct colors for coloring each subset), which achieves the proof. \square

Using Lemma 3.26, Luczak and Thomassé finally broke the $n/3$ threshold using this result (see [140] for a complete proof).

Theorem 3.28 (Luczak, Thomassé [140]). *Every triangle-free graph G with n vertices and minimum degree at least $n/3 - b$ has chromatic number at most $\max(10^5, 12(b + 1))$.*

Generalization: the paired VC-dimension. A class of graphs has *chromatic threshold* 0 if the chromatic number is bounded as long as the minimum degree is at least cn for every positive c . In [140], Luczak and Thomassé studied more precisely other thresholds. They introduced a generalization of VC-dimension, called paired VC-dimension to achieve this goal. A *paired hypergraph* P is a couple (H, G) where H is a hypergraph and G a graph where vertices are hyperedges of H . So edges of G are pairs of hyperedges of H . The *dual paired VC-dimension* is the maximum d such that there exists d edges (E_i, F_i) of G such that for every $I \subseteq \{1, \dots, d\}$, there exists a vertex x such that $x \in E_i$ if $i \in I$ and $x \in F_i$ if $i \notin I$.

paired
hypergraph

dual
paired VC-
dimension

Why did they introduce such an object? The first reason comes from hypergraph theory. One can note that the VC-dimension is not modified when we consider the complement hypergraph, as underlined in Observation 3.3. So much more than the hyperedge e itself, it is the bipartition (e, \bar{e}) which gives information for shattering. The paired VC-dimension is a tool which can break this symmetry. In some sense we can restrict the complement of a hyperedge to a subset of vertices.

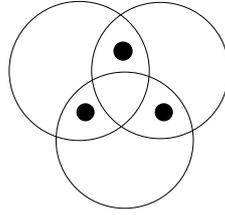


Figure 3.12: A 2-complete Venn diagram of size 3.

The second motivation comes from graph theory. We have already seen that there are many ways to represent graphs as hypergraphs (such as neighborhood hypergraphs). Though in such hypergraphs, we completely forget the original structure of the graph. For instance, in triangle-free graphs, neighborhoods can arbitrarily intersect but neighborhoods of two adjacent vertex are disjoint. So transforming the graph into a hypergraph without keeping in mind the structure of our graph is a loss of information. Transforming it into a paired hypergraph permits to keep some information (in some sense we can still have in mind the edge structure of the original graph).

Theorem 3.29 (Luczak, Thomassé [140]). *There exists a function f such that every paired hypergraph (H, G) of dual paired VC-dimension d satisfies*

$$\chi(G) \leq f(\tau^*(H), d).$$

The quite technical proof is based on an “increasing density argument”. Using Theorem 3.29, Luczak and Thomassé characterized the graphs H such that H -homomorphism-free graphs have chromatic threshold 0. They also provided a sufficient condition in order to have threshold 0 for some H -free graphs generalizing the result of Thomassen on C_{2k+1} -subgraph-free graphs [185]. The reader is referred to [140] for the whole definitions and the proofs.

Clique - Stable set separation problem. Another application of Theorem 3.23 will be provided in Chapter 5 for the so-called clique-stable set separation problem.

3.4 Erdős-Pósa property

In Section 3.3, we stated Theorem 3.22 which ensures that the integrality gap between τ and τ^* is bounded as long as the VC-dimension is bounded. Though, a bounded VC-dimension is not enough to ensure that a class of hypergraphs has the Erdős-Pósa property as underlined in Lemma 3.24. It raises a natural question: can we enforce the definition of VC-dimension in order to bound the gap between τ and ν in the general case. Ding, Seymour and Winkler answered positively to this question by introducing the 2VC-dimension. We will also see in Section 3.4.3 another way to obtain the Erdős-Pósa property using (p, q) -property due to Matoušek.

3.4.1 2VC-dimension

2-shattered

A subset of vertices X of a hypergraph is *2-shattered*, if for every subset $X' \subseteq X$ of size 2, there

exists a hyperedge e such that $e \cap X = X'$. The *2VC-dimension* is the maximum size of a 2-shattered set. The *dual 2VC-dimension* is the maximum size of a 2-shattered set in the dual hypergraph. Note that dual 2VC-dimension is equivalent to the existence of a 2-complete Venn diagram, where 2-complete means that a vertex is contained in all the possible intersections of size 2. In other words a set e_1, \dots, e_d forms a 2-complete Venn diagram if for every i, j there exists $x_{i,j}$ which is in $e_i \cap e_j$ and which is in no other e_k with $k \neq i, j$. Figure 3.12 represents a 2-complete Venn diagram of size 3.

First note that the VC-dimension is not larger than the 2VC-dimension. Indeed, a shattered set is in particular a 2-shattered set. As a consequence, Lemma 3.12 ensures that the maximum number of hyperedges of a hypergraph of 2VC-dimension d is at most $\sum_{i=0}^d \binom{n}{i}$. And Lemma 3.13 ensures that this bound is tight for some hypergraphs. One can naturally ask for the existence of a function linking VC-dimension and 2VC-dimension. The following observation states that such a function does not exist.

Observation 3.30. *For every $n \geq 4$, $\mathcal{U}_{2,n}$ has VC-dimension 2 and 2VC-dimension n .*

Proof. Lemma 3.5, which characterizes the VC-dimension of complete uniform hypergraphs, ensures that the VC-dimension of $\mathcal{U}_{2,n}$ equals 2 (as $\max(2, n-2) = 2$ since $n \geq 4$). And the 2VC-dimension of $\mathcal{U}_{2,n}$ is n since for every pair of vertices, there exists a hyperedge containing both of them and containing no other vertices of the graph. \square

Even more disturbing, an arbitrarily large gap between the 2VC-dimension and the dual 2VC-dimension is possible. The 2VC-dimension is not as stable as the VC-dimension under the dual hypergraph operation. Indeed, Lemma 3.7 ensures that there is a bounded (even if potentially exponential) gap between the VC-dimension and the dual VC-dimension.

Observation 3.31. *For every $n \geq 4$, $\mathcal{U}_{2,n}$ has dual 2VC-dimension 3 and 2VC-dimension n .*

Proof. Observation 3.30 ensures that $\mathcal{U}_{2,n}$ has 2VC-dimension n . Let us show that its dual 2VC-dimension is at most 3. Assume by contradiction that four hyperedges e_1, e_2, e_3, e_4 form a 2-complete Venn diagram. So there are three vertices x_1, x_2, x_3 such that $x_1 \in e_1 \cap e_2 \cap \bar{e}_3 \cap \bar{e}_4$, x_2 is in $e_1 \cap e_3 \cap \bar{e}_2 \cap \bar{e}_4$ and $x_3 \in e_1 \cap e_4 \cap \bar{e}_2 \cap \bar{e}_3$. Hence $|e_1| \geq 3$, a contradiction with the fact that hyperedges of $\mathcal{U}_{2,n}$ have size 2. So the dual VC-dimension of $\mathcal{U}_{2,n}$ is at most 3. \square

The following theorem, due to Ding, Seymour and Winkler is the main result of this Section.

Theorem 3.32 (Ding, Seymour, Winkler [71]). *Every hypergraph H of dual 2VC-dimension d satisfies:*

$$\tau \leq 11d^2(d+v+3) \binom{v+d}{v}^2$$

Hint of proof. We will not provide the whole proof of this statement but in the next few lines, we will explain their method. For more details the reader is referred to [71]. Theorem 2.16 ensures that $\tau \geq \tau^* = v^* \geq v$. The proof roughly consists in proving that both integrality gaps are bounded which is classical method in order to prove the Erdős-Pósa property. Let us detail each step of the proof.

Step 1: Using Theorem 3.23, we have $\tau \leq f(\tau^*, d)$. Indeed since the dual 2VC-dimension is bounded, the dual VC-dimension is bounded. So Lemma 3.7 ensures that VC-dimension is

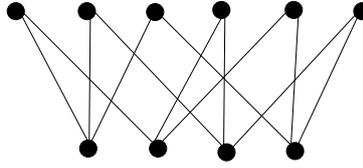


Figure 3.13: A 1-subdivided K_4 . Below stand the “vertices” of the K_4 . Above the are “edges” of the K_4 .

bounded. Finally Theorem 3.23 provides the desired inequality. Their proof is slightly more complicated since such a sequence of inequality does not provide a polynomial gap.

Step 2 The hypergraph duality ensures that $\tau^*(H) = c^*(H^d)$. Recall that c denotes the covering of a hypergraph and c^* its fractional relaxation and α denotes the independent set of a hypergraph. Observation 2.20 ensures $c^*(H^d) = \alpha^*(H^d) = \tau^*(H)$.

Step 3: In the last part, they prove that $\alpha^* \leq f(\alpha, d)$ for every hypergraph. Recall that Ramsey’s Theorem (Theorem 1.3) ensures that every graph contains either a large stable set or a large clique. They prove a generalization of Ramsey’s theorem for hypergraphs which ensures that the hypergraph contains either a 2-shattered subset of vertices or a subset of vertices such that no hyperedge contains more than one vertex in it. Note that in the case of graphs, it provides a clique or a stable set (since no hyperedge has size at least 3). They finally prove that α^* can be bounded by a function of the size of such a set.

A proof of Theorem 3.32 follows since $\tau \leq f(\tau^*, d) \leq g(\alpha(H^d), d) = g(v, d)$. \square

Actually, Ding Seymour and Winkler proved a slightly stronger result than Theorem 3.32. For the sake of completeness, let us state it even if we will not use it in the remaining. A subset of vertices X of a hypergraph is ℓ -shattered, if for every subset $X' \subseteq X$ of size ℓ , there exists a hyperedge e such that $e \cap X = X'$. The ℓ VC-dimension is the maximum size of an ℓ -shattered set. The dual ℓ VC-dimension is the maximum size of a ℓ -shattered set in the dual hypergraph. In other words, we have a vertex in all the possible ℓ -intersections of the set of hyperedges. A generalized ℓ -packing is a set of hyperedges such that no vertex is contained in at least $\ell + 1$ hyperedges. In other words, $\ell + 1$ hyperedges do not intersect on the same vertex. Note that a packing is a generalized 1-packing. Ding, Seymour and Winkler bound the size of a minimum hitting set by a function of the dual ℓ VC-dimension and of the maximum size of a generalized $(\ell - 1)$ -packing.

Theorem 3.32 provides a polynomial gap between τ and v as long as the dual 2VC-dimension is bounded. Nevertheless, nothing was done, up to my knowledge, in order to determine if this bound is optimal or not.

Problem 7. What is the best function f such that $\tau \leq f(v, d)$ where d denotes the 2VC-dimension?

3.4.2 Application: Scott’s Conjecture for maximal triangle-free graphs

The following result is joint work with Stéphan Thomassé and was published in [41]. Recall that χ and ω respectively denote the chromatic number and the size of a maximum clique.

An *induced subdivision* of a graph G is the graph G where each edge is replaced by a path of an arbitrarily length. An *induced 1-subdivision* of a graph G is the graph where every edge of G is subdivided once, *i.e.* is replaced by a path of length 2. In other words, it is an induced bipartite graph $G' = (X, Y)$ such that X has size n and Y has size m and for every edge xx' in G , there exists a vertex y in Y such that $N(y) = \{x, x'\}$. Figure 3.13 represents an induced 1-subdivision of K_4 .

Let F be a graph and $\text{Forb}^*(F)$ be the class of graphs with no induced subdivision of F . A class \mathcal{G} of graphs is χ -bounded if there is a function f such that every graph G of \mathcal{G} satisfies $\chi(G) \leq f(\omega(G))$. Gyárfás proved in [113] that for every (induced) path P_k , $\text{Forb}^*(P_k)$ is χ -bounded. He also conjectured that $\text{Forb}^*(F)$ is χ -bounded if F is a cycle and if F is a tree [113]. Scott proved in [181] that Gyárfás' conjecture holds for trees. And Scott strengthened Gyárfás' conjecture by conjecturing that for every graph F , $\text{Forb}^*(F)$ is χ -bounded. Scott's conjecture also implies a well-known conjecture of Erdős, first cited in [113], stating that every triangle-free intersection graphs of segments in the plane have a bounded chromatic number. Indeed, by planarity, they cannot contain an induced subdivision of a K_5 with all the edges subdivided once. The Erdős' conjecture has recently been disproved by Pawlik et al. in [165]. Nevertheless, the counter-example raises two interesting questions.

 χ -bounded

1. For which graphs F does Scott's conjecture hold? We have already mentioned that it holds for paths and trees. In addition, it holds for every graph on 4 vertices and for bulls (see [57, 141, 186]). In addition, Chalopin et al. [45] determined several graphs F for which Scott's conjecture does not hold using the construction of [165].
2. On which graph classes does Scott's conjecture hold (for every forbidden graph F)? Kuhn and Osthus proved in [131] that Scott's conjecture holds as long as complete bipartite graphs are forbidden. We proved with Stéphan Thomassé that it also holds for maximal triangle-free graphs [41].

Recall that a *maximal triangle-free graph* is a graph G such that for any non edge xy , if xy is added in E , then the graph contains a triangle. More formally, we proved the following:

Theorem 3.33 (B., Thomassé [41]). *There is a constant c such that every maximal triangle-free graph G with $\chi(G) \geq e^{c \ell^4}$ contains an induced subdivision of every graph F on ℓ vertices.*

Proof. The proof consists in showing that if the chromatic number is large enough, then the graph contains a 1-subdivision of K_ℓ . Note that it implies Theorem 3.33 since, given a 1-subdivision of K_ℓ , one can extract a 1-subdivision of any graph of size ℓ . Indeed every edge of K_ℓ is replaced by a path of length 2 in a 1-subdivision. Therefore, by deleting the vertex in the middle of the path, we can eliminate one edge. In other words deleting a vertex of the upper side on Figure 3.13 deletes an edge of the subdivided graph. Let us denote by H the *neighborhood hypergraph* of G , *i.e.* the hypergraph with vertex set V and with hyperedges the closed neighborhoods of the vertices of G .

Observation 3.34. *Every triangle-free graph G with neighborhood hypergraph H satisfies:*

$$\chi(G) \leq 2\tau(H).$$

Proof. The proof looks like the proof of Observation 3.27. Let X be a transversal of the hypergraph H . Every vertex u satisfies $N[u] \cap X \neq \emptyset$, *i.e.* there exists a vertex $x \in X$ such that $u \in N[x]$. So all the

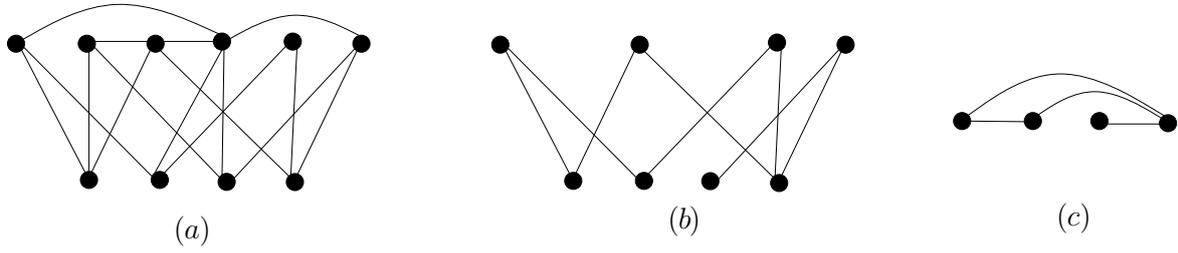


Figure 3.14: Illustration of the proof of Theorem 3.33.

vertices of the graph are covered by the closed neighborhoods of vertices of X . Since G is triangle-free, $N[x]$ is an induced star (a vertex adjacent to a stable set). Hence $N[x]$ can be colored using only 2 colors, i.e. $\chi(G) \leq 2\tau(H)$. \square

Observation 3.35. *Every maximal triangle-free graph satisfies $\nu(H) = 1$.*

Proof. A packing of the neighborhood hypergraph is a subset of vertices such that their closed neighborhoods pairwise do not intersect. If two vertices x and y satisfies $N[x] \cap N[y] = \emptyset$ then adding the edge xy in the graph does not create any triangle, contradicting the assumption that G is a maximal triangle-free graph. So we have $\nu(H) = 1$. \square

Let G be a maximal triangle-free graph of chromatic number at least $e^{c \cdot \ell^4}$. Since $\chi(G) \geq e^{c \cdot \ell^4}$, Observation 3.34 ensures that $\tau \geq 1/2 \cdot e^{c \cdot \ell^4}$. By Observation 3.35, we have $\nu(H) = 1$. So Theorem 3.32 ensures that the dual 2VC-dimension of H is at least $e^{c_2 \cdot \ell^4}$ (for some constant c_2 which depends on c). In other words, there exists a collection of hyperedges e_1, \dots, e_d with $d \geq e^{c_2 \cdot \ell^4}$ which forms a 2-complete Venn diagram. Yet differently, for every pair e_i, e_j there exists a vertex $y_{i,j}$ in $e_i \cap e_j$ which is in no other e_k with $k \neq i, j$. Let us denote by Y the set $\{y_{i,j}, 1 \leq i < j \leq d\}$.

Recall that each e_i corresponds to the closed neighborhood of some vertex x_i of G . Let $X = \{x_1, \dots, x_d\}$ be the subset of vertices corresponding to e_1, \dots, e_d . Since G is triangle-free, Theorem 1.6 ensures that X contains a stable set S of size at least $\sqrt{d} \log d$ (which has size at least $e^{c_3 \cdot \ell^4}$). Replacing X by S , we can assume that X is a stable set. By abuse of notation, we still denote it by $\{x_1, \dots, x_d\}$.

For every i, j , the vertex $y_{i,j}$ is connected to both x_i and x_j and with no other vertex of X . Note that it in particular implies that no $y_{i,j}$ belongs to X since otherwise some $y_{i,j} \in X$ would be a neighbor of another vertex of X , a contradiction since X is a stable set. Consider the subgraph of G induced by the set $X \cup Y$. Figure 3.14(a) illustrates the graph $G[X \cup Y]$.

At this point of the proof we have a (non induced) bipartite graph X, Y where X is a stable set and Y contains $d(d-1)/2$ vertices, each connected to exactly two vertices of X . Note that, if Y is a stable set, $G[X \cup Y]$ is a 1-subdivision of K_d . Indeed, all the $x_i y_{i,j} x_j$ are paths of length 2 between every pair of vertices of X . Nevertheless, Y is not necessarily a stable set and then such a subdivision is not necessarily induced. In the rest of the proof, we show that we can extract from Y a stable set large enough in order to ensure that $G[X \cup Y]$ contains a subdivision of a large clique.

Since the restriction of G to Y is triangle-free (since the whole graph is triangle-free) and since $|Y| = \binom{d}{2}$, Theorem 1.6 ensures that Y contains a stable set Y' of size $\Theta(d\sqrt{\log d})$. Consider the restriction of G to $X \cup Y'$ (see Figure 3.14(b) for an illustration). Construct the graph G' on vertex

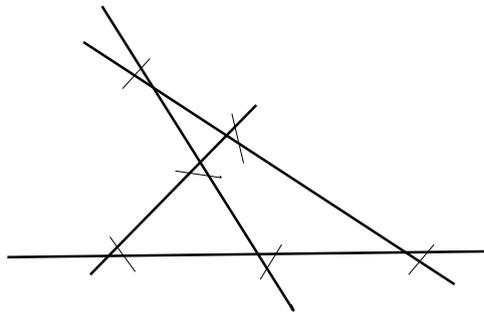


Figure 3.15: The bold segments are 2-shattered in the neighborhood hypergraph. Indeed small segments near the intersection points 2-shatter the four segments.

set X with an edge $x_i x_j$ if and only if $y_{i,j} \in Y'$. The graph G' corresponding to Figure 3.14(b) is represented on Figure 3.14(c). Note that if F is a subgraph of G' (on vertex set X'), then G contains an induced 1-subdivision of F . Indeed, the graph induced by G on $X' \cup Y''$, where $y_{i,j} \in Y''$ whenever $x_i x_j$ is an edge of F , is such a subdivision. Note that since Y' has size $\Theta(d\sqrt{\log d})$, the average degree of G' is $\Theta(\sqrt{\log d})$. So we just have to show that G' contains a subdivision of K_ℓ (and then of F) as a subgraph.

A theorem due to Mader [142] and improved by Bollobás and Thomason [27] ensures that every graph with average degree $512 \cdot \ell^2$ contains a subdivision of K_ℓ . Since $d \geq e^{c_3 \cdot \ell^4}$, there is a constant c_4 such that $\sqrt{\log d} \geq c_4 \ell^2$, hence G' contains a subdivision of K_ℓ , and therefore G has an induced subdivision of K_ℓ (and thus of F). \square

Note that Theorem 3.32 also bounds $\tau(H)$ in terms of $\nu(H)$. Hence the chromatic number of a triangle-free graph G is also bounded when the maximum packing of neighborhoods is bounded. This is the case for instance if the minimum degree is $c \cdot n$ for some fixed constant $c > 0$.

Domination at large distance. We will see in Chapter 4 another applications of Theorem 3.32 for dominating sets at large distance in several classes of graphs, including planar graphs, and more generally K_n -minor-free graphs but also graphs of bounded rankwidth.

3.4.3 VC-dimension and (p,q) -property

There exists another way to obtain the Erdős-Pósa property which is based on the (p,q) -property and VC-dimension. Recall that a hypergraph admits the (p,q) -property if for every set of p hyperedges, at least q of them intersect on the same vertex. Theorem 3.32 provides the Erdős-Pósa property as long as the 2VC-dimension is bounded. Nevertheless, in some cases, the 2VC-dimension can be arbitrarily large and the VC-dimension can be bounded. Such a situation can occur in lots of geometrical objects. Consider for instance the neighborhood hypergraph of graphs of intersection of segments in the plane. Then 2VC-dimension of such hypergraphs is arbitrarily large. Indeed, let X be a set of pairwise disjoint intersecting segments. For every intersection point between two segments of X , add a new segment intersecting only the two segments which intersect (see Figure 3.15 for an illustration). The set X is 2-shattered in the neighborhood hypergraph.

On the contrary, one can prove that the neighborhood hypergraph of such graphs have a bounded VC-dimension (the proof is not immediate). The following theorem due to Matoušek gives other conditions on the hypergraph which ensure that the Erdős-Pósa property holds.

Theorem 3.36 (Matoušek [151]). *Let H be a hypergraph of dual VC-dimension d . There exists a function c such that if H has the $(p, d - 1)$ -property then $\tau \leq c(p, d)$.*

Hint of the proof. The proof follows the scheme of the proof of Alon et Kleitman of the Hadwiger-Debrunner conjecture [8]. Let us informally state the main steps of the proof.

Step 1: Proving that $\tau \leq f(\tau^*, d)$ using Theorem 3.23.

Step 2: $\tau^* = v^*$ by Theorem 2.16.

Step 3: Proving that $v^* \leq f(v)$ using (p, d) -property. As in the Alon-Kleitman proof, the proof is based on a fractional Helly theorem. First prove that every hypergraph of dual VC-dimension at most d has fractional Helly number d . Then, using the Alon-Kleitman proof, we can prove that $v^* \leq f(v, d)$. \square

Theorem 3.36 have been used by Chepoi, Estellon and Vaxès in order to find dominating sets in planar graphs [54]. In Chapter 4, we present joint work with Stéphan Thomassé which generalizes their result.

3.5 Algorithmic aspects of VC-dimension

In this short section, we consider algorithmic aspects of VC-dimension. All along this chapter, we have seen several applications of VC-dimension to obtain some upper bounds on invariants of hypergraphs and in particular to obtain upper bounds on τ . Theorem 3.21 ensures that τ is bounded by a function of τ^* and of the VC-dimension. We have seen that the fractional transversality τ^* can be computed in polynomial time using classical algorithms of linear programming such as the ellipsoid method (Theorem 2.11). So if the VC-dimension can be computed in polynomial time, we can find an algorithm which gives an upper bound on τ in polynomial time. Moreover, we have seen that the proof of Theorem 3.21 provides a (randomized) approximation algorithm. So if the VC-dimension can be easily computed, we would provide immediately an approximation algorithm for finding hitting sets (when the VC-dimension is bounded). Let us now give a formal definition of the decision problem and study its complexity.

VC-DIMENSION

Input: A hypergraph H , an integer k .

Parameter: k

Output: TRUE if the VC-dimension of H is at least k , otherwise FALSE.

Note that the instance is positive if the VC-dimension is at least k and not at most k (as it is usually the case). We do that since the VC-DIMENSION problem is in NP while deciding if a hypergraph has VC-dimension at most k is not clearly in NP. Indeed given a set of size k we can check in polynomial time if it is shattered or not, while determining if a graph has VC-dimension at most k has no (immediate) certificate. Let us first make an observation on the complexity of VC-dimension:

Observation 3.37. *The VC-DIMENSION problem can be solved in $\mathcal{O}^*(n^{\log n})$.*

Proof. The proof of this statement is quite simple. Recall that the VC-dimension of a hypergraph $H = (V, E)$ is at most $\lfloor \ln |E| \rfloor$. Indeed, for every shattered set X , all the possible traces exist on X . Since a set X has $2^{|X|}$ traces, if X is shattered we have $2^{|X|} \leq |E|$.

Hence if k is at least $\lfloor \ln |E| \rfloor + 1$, then we return false. So we can assume that $k \leq \log n$. Consider the naive algorithm trying all the sets of size k . The total number of sets of size k is $\binom{n}{k} \leq n^k$. While $k \leq \log n$, the complexity of the naive algorithm is $\mathcal{O}^*(n^{\log n})$. \square

A problem which can be decided in $\mathcal{O}^*(n^{\mathcal{O}(\log n)})$ is not NP-complete under ETH-conjecture (stating that there is no algorithm for deciding SAT in time $2^{o(n)}$). Indeed assume that an NP-complete problem Π admits a $\mathcal{O}^*(n^{\mathcal{O}(\log n)})$ running time algorithm. Since Π is an NP-complete problem, there exists a polynomial time reduction from SAT to Π . So any instance of SAT of size n is transformed into an instance of Π of size n^c (where c is a fixed constant) which is positive if and only if the original instance is positive. If Π can be decided in $\mathcal{O}^*(n^{\mathcal{O}(\log n)})$, then the initial instance of SAT can be decided in $2^{o(n)}$. A contradiction with the ETH-conjecture.

So, Observation 3.37 ensures that under ETH, the VC-DIMENSION problem is not NP-complete. But on the other side, Downey, Evans and Fellows proved that the VC-DIMENSION problem is $W[1]$ -hard. So if it can be solved in polynomial time (and even on FPT time) then the W -hierarchy collapses. More precisely we have the following:

Theorem 3.38 (Downey, Evans, Fellows [72, 73]). *The VC-DIMENSION problem is $W[1]$ -complete parameterized by k .*

Sketch of the proof. In this proof we do not prove that the VC-DIMENSION problem is in $W[1]$. We refer the reader to [73] for the proof of this part.

Let us prove that the VC-DIMENSION problem is $W[1]$ -hard, a result due to Downey, Evans and Fellows in [72]. The reduction is a reduction from the CLIQUE problem. Recall that for proving that a problem is $W[1]$ -hard we can make reductions in FPT time (*i.e.* non necessarily polynomial in k but polynomial in the rest part of this instance).

Let $G = (V, E)$ be a graph and let k be an integer. In the following we assume that $k \geq 4$. Let us construct hypergraph $H = (X, F)$ which has VC-dimension at least k if and only if G admits a clique of size at least k . The vertex set X of H is composed of all the pairs (v, i) where $v \in V$ and $i \leq k$. Given a vertex (v, i) , the integer i is called the *witness* of the vertex and v is called the *representative* of the vertex. The hyperedge set F is divided into four parts:

- The set F_0 contains only the hyperedge $\{\emptyset\}$.
- The set F_1 contains all the singletons of H , *i.e.* it contains the hyperedges $\{(v, i)\}$ for every $v \in V$ and $i \leq k$.
- The set F_2 contains all the pairs $\{(u, i), (v, j)\}$ where uv is an edge of G and $i, j \leq k$.
- The set F_3 contains all the sets $\{(v, i), \dots, (v, i_{|S|})\}$ for every subset $S \subseteq \{1, \dots, k\}$ of size at least 3. The set S is called the *witness set* of the hyperedge.

Note that this reduction works in FPT time. Indeed F_0, F_1 and F_2 can be constructed in polynomial time. And the construction of F_3 can be done in $\mathcal{O}^*(2^k)$ since we just have to compute all the subsets (of size at least 3) of a set of size k .

Let us show that the VC-dimension of H is least k if and only if G contains a clique of size k . First assume that G contains a clique K of size k . Denote by v_1, \dots, v_k the vertices of K . Let us prove that the set of vertices $Y = \{(v_i, i), i \leq k\}$ is shattered. The set F_0 ensures that there exists a hyperedge

containing no vertex of Y . The set F_1 contains all the hyperedges of size one. Since K induces a clique, the edge $v_i v_j$ exists in G , and then the hyperedge $\{(v_i, i), (v_j, j)\}$ exists in F_2 . So we just have to prove that for every $Y' \subseteq Y$ of size at least 3, there exists a hyperedge e of F_3 such that $e \cap Y = Y'$. Let Y' be a subset of Y . Let S' be the subset of $\{1, \dots, k\}$ such that $i \in S'$ if and only if $v_i \in Y'$. By construction, the hyperedge e of F_3 with witness set S' satisfies $Y \cap e = Y'$. So if G contains a clique of size k then the VC-dimension of H is at least k .

Conversely assume that a set Y of size at least k is shattered. Let us first prove that all the vertices of Y have pairwise distinct witnesses. Assume by contradiction that (u, i) and (v, i) are in Y . Let (w, j) and (x, ℓ) be two other vertices of Y . Such vertices exist since we assumed $k \geq 4$. Let e be a hyperedge containing $(u, i), (w, j)$ and (x, ℓ) . Since $|e| \geq 3$, the hyperedge e is in F_3 . Since $(u, i) \in e$, the witness set S of e contains i . Hence by construction we have $(v, i) \in e$. Thus Y cannot be shattered, a contradiction.

Hence there is a 1 to 1 function between witnesses of vertices of Y and $\{1, \dots, k\}$. Let us now prove that vertices of Y have pairwise distinct representative. Assume by contradiction that (u, i) and (u, j) are in Y . Let e be a hyperedge containing (u, i) and (u, j) . Clearly e is not in F_0, F_1 and F_2 , so $e \in F_3$. Denote by S' the witness set of e . Let ℓ be in S' such that $\ell \neq i, j$. Such a ℓ exists since S' has size at least 3. Since there is a bijection between $\{1, \dots, k\}$ and witnesses, there exists a vertex y of Y which has witness ℓ . So $e \cap Y$ also contains y , *i.e.* the set Y cannot be shattered, a contradiction.

Finally a shattered set is a set (v_i, i) for every $i \leq k$ where the vertices v_i are pairwise distinct. In order to shattered sets of size 2, we have to use hyperedges of F_2 and then for every $i \neq j$, we have $v_i v_j \in E$. So the set of representative of Y induces a clique in G . \square

Note that the reduction of Theorem 3.38 is not polynomial. And it is quite comforting since otherwise it would in particular prove that the VC-DIMENSION problem is NP-complete which would contradict the ETH because of Observation 3.37.

Finally the VC-DIMENSION problem seems to not be polynomial time solvable because of Theorem 3.38 and seems to not be NP-complete because of Observation 3.37. So we can naturally think that the VC-DIMENSION problem is a problem which is in a class between P and NP . Papadimitriou and Yannakakis introduced in [164] a new class of complexity which is between P and NP and they proved that the VC-DIMENSION problem is complete for this class.

Domination at large distance

In [54], Chepoi, Estellon and Vaxès proved that every planar graph of diameter 2ℓ can be covered by c balls of radius ℓ (where c is independent from the graph and from ℓ). In other words, the class of hypergraphs of balls of radius ℓ of planar graphs of diameter 2ℓ has the Erdős-Pósa property. Their proof is based on Theorem 3.36 and on topological properties of planar graphs. Our initial motivation when we look at this problem was to determine if we can avoid the part using topological property and just obtain a combinatorial proof. This chapter is derived from a paper of Stéphan Thomassé and myself in which:

- We first define notions of VC-dimension and 2VC-dimension on graphs and prove that clique-minor free graphs and bounded rankwidth graphs have bounded 2VC-dimension.
- We then give a short proof of the result of Chepoi, Estellon and Vaxès based on Theorem 3.32.
- We finally show that the hypergraph of balls of radius ℓ of any graph of bounded VC-dimension satisfies the Erdős-Pósa property (where the function does not depend on ℓ but only on v and the VC-dimension) using Theorem 3.36

4.1 Introduction

We have seen in Chapter 2 and 3 that there are several ways to represent a graph as a hypergraph. In most of the applications of Chapter 3, we consider the neighborhood hypergraph. Nevertheless this hypergraph cannot catch the whole complexity of a graph. For instance bounded degree graphs have “simple” neighborhood hypergraphs (since the sizes of the hyperedges are bounded) but they can have a random behavior at large distance (think about random cubic graphs).

Recall that the B -hypergraph of a graph G has vertex set V and that a subset $Y \subseteq V$ is a hyperedge if there are a vertex $x \in V$ and an integer k such that $Y = B(x, k)$. For a given integer ℓ , the B_ℓ -hypergraph of G has vertex set V and $Y \subseteq V$ is a hyperedge if there is an x such that $Y = B(x, \ell)$ (where $B(x, \ell)$ is the set of vertices at distance at most ℓ from x).

The VC-dimension of a graph. The VC-dimension of a graph G could be defined as the VC-dimension of the B -hypergraph of the graph G . Though, in order to ensure some stability, we close

this notion by induced subgraphs. So the *VC-dimension of a graph G* is the maximum by all induced subgraphs of the VC-dimension of the B -hypergraph. Let us first make the following observation.

Observation 4.1. *The VC-dimension of a non connected graph is the maximum of the VC-dimension of its connected components.*

Since the VC-dimension “measures” the randomness of hypergraphs, it is natural to think that classes with lots of structure can have a bounded VC-dimension. Note that since we consider iterated neighborhoods and not only neighborhoods (at distance one), we can catch the “randomness at large distance”. For instance, with our definition, random cubic graphs have an unbounded VC-dimension while their neighborhood hypergraphs have VC-dimension at most 3.

In Section 4.2, we prove that two famous graph classes have bounded VC-dimension. First we show that the class of K_n -minor free graphs has VC-dimension at most $n - 1$. The proof is almost the proof of Chepoi, Estellon and Vaxès that the B_ℓ -hypergraph of planar graphs has VC-dimension at most 4. Then we show that the class of bounded rankwidth graphs have bounded VC-dimension. Actually, we prove a slightly stronger statement for these two classes: their 2VC-dimension is bounded. We finally provide some graphs of bounded VC-dimension and with an arbitrarily large 2VC-dimension.

Erdős-Pósa property. In Section 4.3, we look for the Erdős-Pósa property of the B_ℓ -hypergraphs for graphs of bounded VC-dimension. Chepoi, Estellon and Vaxès [54] proved that every planar graph of diameter 2ℓ can be covered by c balls of radius ℓ (where c does not depend on ℓ). It answers a conjecture of Gavaille, Peleg, Raspaud and Sopena [105]. Their proof is based on both combinatorial arguments (based on VC-dimension) and topological arguments (based on planar graphs). Recall that this result can be seen as an Erdős-Pósa property since every pair of balls of radius ℓ intersects in a graph of diameter 2ℓ . We will see a little bit further that combinatorial assumptions are enough for obtaining equivalent results. We say that a set X is a *dominating set at distance ℓ* if the set of balls of radius ℓ centered in X covers all the vertices of the graph. We denote by ν_ℓ and τ_ℓ , respectively, the packing number and the transversality of the B_ℓ -hypergraph of G . Let us first make a simple observation.

Observation 4.2. *The B_ℓ -hypergraph is isomorphic its dual.*

Proof. For every pair x, y of vertices, we have $x \in B(y, \ell)$ if and only if $y \in B(x, \ell)$. So x is in the hyperedge corresponding to y if and only if y is in the hyperedge corresponding to x . It ensures that the dual hypergraph is exactly the primal one. \square

Observation 2.9 ensures the dual of a hitting set is a covering. A covering is a subset of hyperedges such that every vertex is in at least one hyperedge. So, in the B_ℓ -hypergraph, a covering is a subset of vertices X such that the balls centered in X of radius ℓ cover the vertices of the graph. Finally Lemma 2.9 and Observation 4.2 ensures the following.

Corollary 4.3. *Let G be a graph. A hitting set of the B_ℓ -hypergraph of G is a dominating set at distance ℓ of G .*

So finding a dominating set at distance ℓ corresponds to finding a hitting set of the B_ℓ -hypergraph. In the following, we will only consider this problem from a hitting set point of view.

First we simplify (and generalize) the proof of Chepoi, Estellon and Vaxès using Theorem 3.32. More precisely we prove that the B_ℓ hypergraph of any graph G has transversality at most $\mathcal{O}(v_\ell^{2d+1})$ where d denotes the 2VC-dimension of G . Note that the function does not depend on ℓ but only on v_ℓ . Since planar graphs have VC-dimension at most 4, it ensures that the B_ℓ hypergraph of any planar graph of diameter 2ℓ satisfies $\tau_\ell \leq 88000$.

Since some graphs of bounded VC-dimension have an arbitrarily large 2VC-dimension, it raises a natural question: is it possible to extend the result which holds for graphs of bounded 2VC-dimension to graphs of bounded VC-dimension. The main part of this chapter consists in proving that the answer to this question is positive. More formally, we prove that there exists a function f such that the B_ℓ hypergraph of a graph of VC-dimension d has a hitting set of size at most $f(v_\ell, d)$. Our proof is based on Theorem 3.36. The original proof of Chepoi, Estellon and Vaxès uses the same argument but they conclude using topological properties of planar graphs. Since we only deal with combinatorial structure, our proof is more technically involved.

In our proof of the Erdős-Pósa property, we do not make any attempt to improve the gap function. We need an exponential gap function at many steps, for instance we make several Ramsey's extractions and we use Theorem 3.36 whose bound is not polynomial also. Finding a polynomial gap function instead of an exponential one is an interesting, but also probably really complicated problem. We can also restrict this questions for several classes of graphs. For instance, Chepoi, Estellon and Vaxès conjectured that the gap function linking v_ℓ and τ_ℓ for planar graphs is a linear function. More formally they conjectured the following.

Conjecture 6. (Chepoi, Estellon, Vaxès [54]) *There exists a constant c such that $\tau_\ell(G) \leq c \cdot v_\ell(G)$ for every ℓ and every planar graph G .*

Note also that Dvorak proved in [77] that there exists a polynomial function P such that $\tau_\ell \leq \text{Poly}(\ell, v_\ell)$ for bounded expansion classes. Note that this proof is not based on VC-dimension. Nevertheless, this polynomial function depends on ℓ while it is not the case in our case nor in the proof of Chepoi, Estellon and Vaxès.

4.2 Graphs of bounded VC-dimension

In this section prove that some graph classes have bounded VC-dimension. More precisely, minor-free graphs and bounded rank-width graphs have bounded 2VC-dimension. In addition we provide a class of graphs with arbitrarily large 2VC-dimension with VC-dimension at most 18.

4.2.1 K_d -minor-free graphs have bounded VC-dimension

A graph H is a *minor* of G if H can be obtained from G by contracting edges, deleting edges, and deleting vertices. Theorem 4.5 is roughly Proposition 1 of [54]. Since our definitions and statements are slightly different, we prove it for the sake of completeness. We first prove an easy lemma before stating the main theorem of this section.

Lemma 4.4. *If z is on a minimum xy -path, $B(z, d(x, z)) \subseteq B(y, d(x, y))$.*

Proof. Since z is on a minimum xy -path, $d(x, y) = d(x, z) + d(z, y)$. Hence $B(y, d(y, z))$ contains z and then $B(y, d(y, z) + d(z, x))$ contains $B(z, d(x, z))$. \square

Theorem 4.5. *A K_d -minor-free graph has 2VC-dimension at most $d - 1$.*

Proof. Let G be a graph with 2VC-dimension d . Let $X = \{x_1, x_2, \dots, x_d\}$ be a set of vertices of G which is 2-shattered by the hyperedges of the B -hypergraph of G . Hence, for every pair (i, j) , there exists a vertex $c_{i,j}$ and an integer $r_{i,j}$ such that $B(c_{i,j}, r_{i,j}) \cap X = \{x_i, x_j\}$, we assume moreover that $r_{i,j}$ is minimum for all choices of $(c_{i,j}, r_{i,j})$. A *central path* $P_{i,j}$ is the concatenation of a minimum path from x_i to $c_{i,j}$ and a minimum path from $c_{i,j}$ to x_j .

Claim 4.6. *A central path is indeed a path.*

Proof. Assume by contradiction that x appears more than once in a central path $P_{i,j}$. Since $P_{i,j}$ is a concatenation of a $x_i c_{i,j}$ -path and a $c_{i,j} x_j$ -path, x appears once between x_i and $c_{i,j}$ and once between $c_{i,j}$ and x_j . Let us call Q_1 the subpath of $P_{i,j}$ from x to $c_{i,j}$ and Q_2 the subpath of $P_{i,j}$ from $c_{i,j}$ to x . Note that Q_1 and Q_2 are both minimum paths connecting $c_{i,j}$ and x , hence replacing Q_2 by the mirror of Q_1 gives another central path $P'_{i,j}$. The two neighbors of $c_{i,j}$ in $P'_{i,j}$ are the same vertex v , contradicting the minimality of $r_{i,j}$ since $B(v, r_{i,j} - 1) \cap X = \{x_i, x_j\}$. \square

Claim 4.7. *If x belongs to two distinct central paths, then these paths are $P_{i,j}$ and $P_{i,l}$, and we both have $d(x, x_i) < d(x, x_j)$ and $d(x, x_i) < d(x, x_l)$.*

Proof. Assume that x appears in $P_{i,j}$ and $P_{k,l}$, where $d(x, x_i) \leq d(x, x_j)$ and $d(x, x_k) \leq d(x, x_l)$. Free to exchange the roles of $P_{i,j}$ and $P_{k,l}$, we can also assume that $d(x, x_k) \leq d(x, x_i)$. By Lemma 4.4, $x_k \in B(c_{i,j}, r_{i,j})$, hence we have $x_k = x_i$ or $x_k = x_j$. Since $d(x, x_k) \leq d(x, x_i) \leq d(x, x_j)$ and x_k is either x_i or x_j , we have $d(x, x_k) = d(x, x_i)$. Hence $d(x, x_i) \leq d(x, x_k)$, and by the same argument, we have $x_i = x_k$ or $x_i = x_l$. Since the central paths are distinct, we necessarily have $x_i = x_k$. Observe that $d(x, x_i) = d(x, x_j)$, hence $d(x, x_j) \leq d(x, x_i)$, would give by the same argument $x_j = x_k$, hence a contradiction since we would have $x_i = x_j$. Therefore $d(x, x_i) < d(x, x_j)$, and for the same reason $d(x, x_i) < d(x, x_l)$. \square

Let us now construct some connected subsets X_i for all $1 \leq i \leq d$. For every path $P_{i,j}$, the vertices of $P_{i,j}$ closer to x_i than to x_j are added to X_i , the vertices of $P_{i,j}$ closer to x_j than to x_i are added to X_j , and the midvertex (if any) is arbitrarily added to X_i or to X_j .

The crucial fact is that the sets X_i are pairwise disjoint. Indeed, by Claim 4.6 and Claim 4.7, if a vertex x appears in two distinct central paths, these are $P_{i,j}$ and $P_{i,l}$, where $d(x, x_i) < d(x, x_j)$ and $d(x, x_i) < d(x, x_l)$. In particular x is added in both cases to X_i .

By construction, the sets X_i are connected and there is always an edge between X_i and X_j since their union contains $P_{i,j}$. Therefore if the 2VC-dimension is at least d , the graph contains K_d as a minor. \square

4.2.2 Bounded rankwidth graphs have bounded VC-dimension

Let us first recall the definition of rankwidth, introduced by Oum and Seymour in [159]. Let $G = (V, E)$ be a graph and (V_1, V_2) be a partition of V . Let M_{V_1, V_2} be the matrix of size $|V_1| \times |V_2|$ such that the entry $(x_1, x_2) \in V_1 \times V_2$ equals 1 if $x_1 x_2 \in E$ and 0 otherwise. The *cutrank* $cr(V_1, V_2)$ of (V_1, V_2)

is the rank of the matrix M_{V_1, V_2} over the field \mathbb{F}_2 . A *ternary tree* is a tree with nodes of degree 3 or 1. The nodes of degree 3 are the *internal nodes*, the other nodes being the *leaves*. A *tree-representation of G* is a pair (T, f) where T is a ternary tree with $|V|$ leaves and f is a bijection from V to the set of leaves. Every edge e of T defines a partition of the leaves of T . Therefore it defines a partition of the vertex set V into (V_1^e, V_2^e) . The *rankwidth rw* of a graph G is defined by:

$$rw(G) = \min_{(T, f)} \max_{e \in E(T)} cr(V_1^e, V_2^e)$$

Before stating the main result, let us first recall two well-known statements concerning rankwidth and ternary trees.

Lemma 4.8. *Let $G = (V, E)$ be a graph of rankwidth k and X, Y be the partition of V induced by an edge of a tree-representation of G of cutrank k . There exist partitions of X and Y into 2^k sets X_1, \dots, X_{2^k} and Y_1, \dots, Y_{2^k} such that for all i, j , $(X_i \times Y_j) \cap E = \emptyset$ or $(X_i \times Y_j) \cap E = X_i \times Y_j$.*

Proof. Let T be a tree representation of G of cutrank at most k . Let e be an edge of the tree representation of G and (X, Y) be the partition of V induced by e . Since the cutrank is at most k , the matrix $M_{X, Y}$ has rank at most k . Hence k rows R_1, \dots, R_k form a base of the rows of the matrix $M_{X, Y}$. By definition, every row corresponds to the neighborhood of a vertex of X into Y . Let us denote by x_i the vertex corresponding to R_i . We denote by \mathcal{B} the set $\{x_1, \dots, x_k\}$.

For every $\mathcal{B}' \subseteq \mathcal{B}$, $X(\mathcal{B}')$ denotes the subset of X which contains x if $N(x) \cap Y =_{\mathbb{F}_2} \sum_{x_i \in \mathcal{B}'} N(x_i)$. It induces a partition of X since $N(x_1), \dots, N(x_k)$ is a base of the neighborhoods of X in Y . Note that by definition all the vertices of $X(\mathcal{B}')$ have the same neighborhood in Y . Observe that a vertex $x \in X(\mathcal{B}')$ is connected to a vertex y if and only if an odd number of vertices of \mathcal{B}' are connected to y .

For every $\mathcal{B}' \subseteq \mathcal{B}$, $Y(\mathcal{B}')$ is the subset of Y containing y if $N(y) \cap \mathcal{B} = \mathcal{B}'$. It induces a partition of Y into 2^k sets with the same neighborhood in \mathcal{B} .

Let us finally prove that the partitions of $X(\mathcal{B}')_{\mathcal{B}' \subseteq \mathcal{B}}$ and $Y(\mathcal{B}')_{\mathcal{B}' \subseteq \mathcal{B}}$ satisfy the required properties. Let x, y be in $X(\mathcal{B}') \times Y(\mathcal{B}'')$ such that xy is an edge. Since xy is an edge, an odd number of vertices of \mathcal{B}' are connected to y . Since all the vertices of $Y(\mathcal{B}'')$ have the same neighborhood in \mathcal{B} , all the vertices of $Y(\mathcal{B}'')$ have an odd number of neighbors on \mathcal{B}' . Thus x is connected to all the vertices of $Y(\mathcal{B}'')$. Since all the vertices of $X(\mathcal{B}')$ have the same neighborhood in Y , $(X(\mathcal{B}'), Y(\mathcal{B}''))$ forms a complete bipartite graph. \square

Lemma 4.9. *Every ternary tree T with $\alpha > 2$ labeled leaves has an edge e such that the partition induced by e has at least $\alpha/3$ labeled leaves in both of its two connected components.*

Proof. Orient every edge of T from the component with less labeled leaves to the other one (when equality holds, orient arbitrarily). Observe that leaves are sources of this oriented tree. Let v be an internal node of T which is a sink. Consider a component C of $T \setminus v$ with at least $\alpha/3$ labeled leaves. Call $e = vw$ the edge of T inducing the partition $(T \setminus C, C)$. Since e is oriented from w to v , the component $T \setminus C$ has at least $\alpha/2$ labeled leaves, thus e is the edge we are looking for. \square

Theorem 4.10. *The 2VC-dimension of a graph with rankwidth k is at most $3 \cdot 2^{k+1} + 2$.*

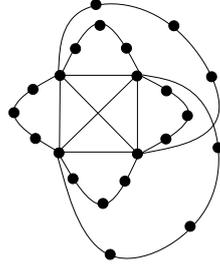


Figure 4.1: The graph $G_{n,\ell}$ of Theorem 4.12 with $n = 4$ and $\ell = 2$.

Proof. Assume by contradiction that the B -hypergraph of a graph G of rankwidth k admits a 2-shattered set S of size $3(2^{k+1} + 1)$. Let (T, f) be a tree decomposition of G achieving rankwidth k . By Lemma 4.9, there is an edge e of T such that the partition induced by e has at least $2^{k+1} + 1$ vertices of S in both connected components. Let V_1, V_2 (resp. X, Y) be the partition of V (resp. S) induced by e . Let $x_1, \dots, x_{2^{k+1}+1}$ and $y_1, \dots, y_{2^{k+1}+1}$ be distinct vertices of X and Y respectively.

Since S is 2-shattered, for each $(x_i, y_j) \in X \times Y$, there is a ball $B(i, j)$ such that $B(i, j) \cap S = \{x_i, y_j\}$ where $B(i, j)$ is chosen with minimum radius.

Claim 4.11. *One of the following holds:*

- There is an i such that at least $2^k + 1$ balls $B(i, j)$ have their centers in V_1 .
- There is a j such that at least $2^k + 1$ balls $B(i, j)$ have their centers in V_2 .

Proof. Orient the edges of the complete bipartite graph with vertex set $X \cup Y$ such that $x_i \rightarrow y_j$ if $B(i, j)$ has its center in V_1 and $x_i \leftarrow y_j$ otherwise. The average out-degree of the vertices of $X \cup Y$ is $2^k + 1/2$, thus there is a vertex with out-degree at least $2^k + 1$, i.e. a vertex satisfying one of the conditions of Claim 4.11. \square

By Claim 4.11, we can assume without loss of generality that $B(1, 1), B(1, 2), \dots, B(1, 2^k + 1)$ have their centers in V_1 . We denote by c_i and r_i respectively the center and the radius of $B(1, i)$ and by P_i a minimum $c_i y_i$ -path. By the pigeonhole principle, two P_i 's leave V_1 by the same set of vertices given by the partition of Lemma 4.8. Without loss of generality, we assume that these paths are P_1 and P_2 and we denote by z_1 and z_2 respectively their last vertices in V_1 . We finally assume that $d(z_1, y_1) \leq d(z_2, y_2)$. By Lemma 4.4, $B(z_2, d(z_2, y_2)) \subseteq B(c_2, r_2)$ since z_2 is on a minimum path from c_2 to y_2 . Let $z_1 z'_1$ be the first edge of P_1 between z_1 and y_1 (hence z'_1 belongs to Y). By Lemma 4.8, z'_1 is also a neighbor of z_2 since z_1 and z_2 have the same neighborhood in Y . Thus $y_1 \in B(z_2, d(z_2, y_2))$. Thus $y_1 \in B(z_2, d(z_2, y_2))$ which contradicts the hypothesis. \square

Since the rankwidth is equivalent, up to an exponential function, to the cliquewidth, Theorem 4.10 implies that the class of bounded clique-width graphs has bounded 2VC-dimension.

4.2.3 Unbounded 2VC-dimension but bounded VC-dimension

Theorem 4.12. *Let n, ℓ be two integers. There exists a graph $G_{n,\ell}$ of VC-dimension at most 18 such that the 2VC-dimension of the B_ℓ -hypergraph of $G_{n,\ell}$ is at least n .*

Proof. The following construction is illustrated on Figure 4.1. The graph $G_{n,\ell}$ has vertex set $X \cup Y$. The set X contains n vertices denoted by $(x_i)_{1 \leq i \leq n}$ and Y is a set of $(2\ell - 1) \binom{n}{2}$ vertices denoted by $y_k^{i,j}$ where $1 \leq k \leq 2\ell - 1$ and $1 \leq i < j \leq n$. The graph restricted to X is a clique. The graph restricted to Y is a disjoint union of $\binom{n}{2}$ induced paths on $2\ell - 1$ vertices (whose endpoints will be connected to vertices of X). More formally, for every $1 \leq i < j \leq n$ and $k \leq 2\ell - 1$, the neighbors of the vertex $y_k^{i,j}$ are the vertices $y_{k-1}^{i,j}$ and $y_{k+1}^{i,j}$ where $y_0^{i,j}$ is x_i and $y_{2\ell}^{i,j}$ is x_j . For every $i < j$, the path $x_i, y_1^{i,j}, y_2^{i,j}, \dots, y_{2\ell-1}^{i,j}, x_j$ is called the *long path between x_i and x_j* .

The 2VC-dimension of the B_ℓ -hypergraph of $G_{n,\ell}$ is at least n . Indeed the vertices of X are 2-shattered since for every $x_i, x_j \in X$, we have $B(y_1^{i,j}, \ell) \cap X = \{x_i, x_j\}$. The rest of the proof consists in proving that the VC-dimension of $G_{n,\ell}$ is at most 18. Consider an induced subgraph of $G_{n,\ell}$. All the remaining vertices of X are in the same connected component since X is a clique. Connected components which do not contain vertices of X form an induced path and then have VC-dimension at most two by Theorem 4.5. So Observation 4.1 ensures that Theorem 4.12 holds if it holds for the connected component of X .

Claim 4.13. *A shattered set of size at least four cannot contain 3 vertices on the same long path.*

Proof. Let z_1, z_2, z_3 be three vertices on the same long path P and z_4 be a vertex which is not between z_1 and z_3 on P . By construction, every path between z_2 and z_4 intersects either z_1 or z_3 . So no pair $z, p \in V \times \mathbb{N}$ satisfy $B(z, p) \cap X = \{z_2, z_4\}$, i.e. $\{z_1, z_2, z_3, z_4\}$ is not shattered. \square

Let Z' be a shattered set of size at least 19. By Claim 4.13, we can extract from Z' a set Z of size 10 such that vertices of Z are in pairwise distinct long paths. For every vertex $z_i \in Z$, a *nearest neighbor* on X is a vertex x of X such that $d(x, z_i)$ is minimum. Note that each vertex has at most two nearest neighbors which are the endpoints of the long path containing z_i .

First assume z_1, z_2, z_3 in Z have a common nearest neighbor x , i.e. they are on long paths containing x as endpoint. Without loss of generality $d(z_3, X)$ is minimum. Let z, p be such that $\{z_1, z_2\} \subseteq B(z, p)$. Free to exchange z_1 and z_2 , a minimum $z z_2$ -path passes through a vertex y of X since z_1 and z_2 are not in the same long path. If $y = x$, then $B(z, p)$ contains $B(x, d(x, z_2))$ by Lemma 4.4, and then contains z_3 since $d(x, z_2) \geq d(x, z_3)$. Otherwise up to symmetry y is not an endpoint of the long path of z_2 . Indeed the second endpoints of the long path containing z_1 and of the long path containing z_2 are distinct (otherwise they would be in the same long path). Hence $d(y, z_2) \geq d(y, z_3)$. So z_3 is in $B(z, p)$ and $\{z_1, z_2, z_3\}$ cannot be shattered.

So each vertex of Z has at most two nearest neighbors in X and each vertex of X is the nearest neighbor of at most two vertices of Z . Thus every $z \in Z$ share a common nearest neighbor with at most two vertices of Z . Since $|Z| \geq 10$, at least four vertices z_1, z_2, z_3, z_4 of Z have distinct nearest neighbors. Assume w.l.o.g. that $d(z_4, X)$ is minimum.

Let $z, p \in V \times \mathbb{N}$ be such that $B(z, p)$ contains z_1, z_2, z_3 . Free to permute z_1, z_2, z_3 , the nearest neighbors of z are not nearest neighbors of z_3 (since z has at most 2 nearest neighbors and nearest neighbors of z_1, z_2, z_3 are pairwise distinct). So a minimum path from z to z_3 passes through a vertex $x \in X$ such that $d(x, z_3) \geq d(x, z_4)$. By Lemma 4.4, $B(z, p)$ also contains z_4 , i.e. Z is not shattered. \square

Note that we did not make any attempt in order to exactly evaluate the VC-dimension of the graph $G_{n,\ell}$.

4.3 Erdős-Pósa property

Recall that ν_ℓ and τ_ℓ respectively denote the packing number and the transversality of the B_ℓ -hypergraph of G . Chepoi, Estellon and Vaxès proved in [54] that there is a constant c such that for all ℓ , every planar graph G of diameter 2ℓ can be covered by c balls of radius ℓ . It means that planar graphs of diameter 2ℓ satisfy $\tau_\ell \leq f(\nu_\ell)$: indeed two balls of radius ℓ necessarily intersect. They raised the following question. Does there exist a function f such that $\tau_\ell \leq f(\nu_\ell)$ for every ℓ and every planar graph (with no constraint on the diameter)? Theorem 3.32 permits to answer positively to this question for graphs of bounded 2VC-dimension.

Corollary 4.14. *Let d be an integer. For every graph $G \in \mathcal{G}$ and every integer ℓ , if the 2VC-dimension of G is at most d , then*

$$\tau_\ell \leq 11 \cdot d^2 \cdot (d + \nu_\ell + 3) \cdot \binom{d + \nu_\ell}{d}^2$$

Proof. Let G be a graph. Observation 4.2 ensures that the B_ℓ -hypergraph of G is isomorphic its dual hypergraph. And the B_ℓ -hypergraph of G is a sub-hypergraph (in sense of hyperedges) of the B -hypergraph of G . Hence the dual 2VC-dimension of the B_ℓ -hypergraph of G is at most d and then Theorem 3.32 can be applied. \square

Theorems 4.5 and 4.10 and Corollary 4.14 ensure that B_ℓ -hypergraphs of K_n -minor free graphs and of bounded rankwidth graphs have the Erdős-Pósa property. Note that the gap between ν_ℓ and τ_ℓ is a polynomial function when the VC-dimension is fixed. Since Theorem 4.12 ensures that there are some graphs with bounded VC-dimension and unbounded 2VC-dimension, Corollary 4.14 raises a natural question. Does the same hold for graphs of bounded VC-dimension? The rest of this Section is devoted to answering this question.

Theorem 4.15. *There exists a function f such that, for every ℓ , every graph of VC-dimension d can be covered by $f(\nu_\ell, d)$ balls of radius ℓ , i.e. $\tau_\ell \leq f(\nu_\ell, d)$.*

Our proof follows the scheme of the proof of Chepoi et al. [54]: both are based on a result of Matoušek linking (p, q) -property and Erdős-Pósa property [151]. Nevertheless our proof is more technical since we cannot use topological properties as for planar graphs in [54].

A hypergraph has the (p, q) -property if for every set of p hyperedges, q of them have a non-empty intersection, i.e. there is a vertex v in at least q of the p hyperedges. The following result, due to Matoušek [151], generalizes a result of Alon and Kleitman [8].

Theorem 4.16. *(Matoušek [151]) There exists a function f such that every hypergraph H of dual VC-dimension d satisfying the $(p, d-1)$ -property satisfies*

$$\tau(H) \leq f(p, d)$$

Let d be an integer. Let G be a graph of VC-dimension d . By Observation 4.2, the dual VC-dimension of the B_ℓ -hypergraph is at most d . Hence if there exists a function p such that, for every ℓ and every graph G of VC-dimension d , the B_ℓ -hypergraph of G satisfies the $(p(\nu_\ell, d), d-1)$ -property, then Theorem 4.16 will ensure that Theorem 4.15 holds.

In the following, we prove that the size of a set of balls of radius ℓ which does not contain $d - 1$ balls intersecting on a same vertex is bounded by a function of v_ℓ and d . The proof is structured as follows: first we prove a general theorem giving lower bounds on VC-dimension of graphs and we give a first application of it. We also provide an important lemma on the structure of the edges between minimum paths. We then try to define a notion of independence between minimum paths. Our goal is to prove that the VC-dimension is a function of the number of “independent” paths.

4.3.1 A lower bound for the VC-dimension of a graph

Let A and B be two disjoint sets. An *interference matrix* $M = (A, B)$ is a matrix with $|A|$ rows and $|B|$ columns such that for every $(a, b) \in A \times B$, the *entry* $m(a, b)$ is a subset of $(A \cup B) \setminus \{a, b\}$. The *size* of an entry is its number of elements. A k -*interference matrix* M is an interference matrix which entries have size at most k . If $A' \subseteq A$ and $B' \subseteq B$, the *submatrix* M' of M induced by $A' \times B'$ is the matrix restricted to the set of rows A' and the set of columns B' which entries are $m'(a', b') = m(a', b') \cap (A' \cup B')$. A 0-interference matrix is called a *proper matrix*. A matrix is *square* if $|A| = |B|$. The *size* of a square matrix is its number of rows.

Lemma 4.17. *Let $k > 0$. A k -interference square matrix with no proper submatrix of size n has size at most kn^3 .*

Proof. Let us show that if $M = (A, B)$ is a k -interference matrix with size $m = kn^3 + 1$, then it contains a proper submatrix of size n . A triple $(i, j, l) \in A \times B \times (A \cup B)$ is a *bad triple* if $l \in m(i, j)$ and $l \neq i$ and $l \neq j$. Let $X \subseteq A$ and $Y \subseteq B$ be two subsets of size n chosen uniformly at random. Let us denote by $1_{i,j,l}$ the random variable which is equal to 1 if $\{i, j, l\} \subseteq X \cup Y$ and 0 otherwise. The expected value of $1_{i,j,l}$ is given by:

$$\mathbb{E}(1_{i,j,l}) = n/m \cdot n/m \cdot (n-1)/(m-1)$$

The third term of the product follows from the fact that l is neither i or j . Since the number of bad triples is at most k for each entry, there are at most km^2 bad triples. Hence the expected number of bad triples is at most $kn^2(n-1)/(m-1) = 1 - 1/n$ which is less than one. Hence there is a pair (X, Y) for which none of the bad events happen. The restriction of M to (X, Y) gives a proper submatrix of size n . \square

Given a path P from x to y and a path Q from y to z , the *concatenation* of P and Q denoted by PQ is the walk consisting on the edges of P followed by the edges of Q . The length of a path P is denoted by $|P|$. Let $G = (V, E)$ be a graph and $<_l$ be a total order on E . We extend $<_l$ on paths, for any paths P_1 and P_2 as follows :

- If P_1 has no edges, then $P_1 <_l P_2$.
- If $P_1 = P'_1 \cdot e_1$ and $P_2 = P'_2 \cdot e_1$, where e_1 is the last edge of P_1 and P_2 , then $P_1 <_l P_2$ if and only if $P'_1 <_l P'_2$.
- If $P_1 = P'_1 \cdot e_1$ and $P_2 = P'_2 \cdot e_2$, where $e_1 \neq e_2$, then $P_1 <_l P_2$ if and only if $e_1 <_l e_2$.

The order $<_l$ is called the *lexicographic order* (note nevertheless that paths are compared from their end to their beginning). The *minimum path* from x to z , denoted by *the xz -path*, is the path of minimum length with minimum lexicographic order from x to z . Observe that two minimum paths going to the same vertex z and passing through the same vertex u coincide between u and z . We note $u \triangleleft_{xz} v$ if u appears before v on the xz -path. Given a path from a to b passing through c , the

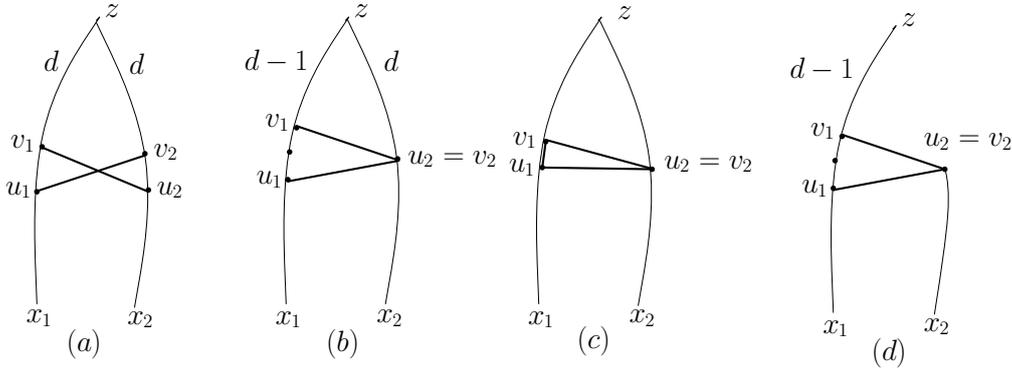


Figure 4.2: Lemma 4.18 ensures that, up to symmetry, only (c) and (d) are authorized. The thick chords are edges of the graph. Thin chords represent paths. Distances are denoted by d or $d - 1$. In the case of Figure 4.2(d), the path Q_{v_2} is $v_2 v_1 Q_{v_1}$.

suffix path on c (resp. *prefix path on c*) is the cb -subpath (resp. ac -subpath) of the ab -path. Note that every suffix of a minimum path is a minimum path. Given two sets X and Z , the XZ -paths are the xz -paths for all $x, z \in X \times Z$.

Let x_1, x_2 and z be three vertices. Two distinct edges $v_1 u_2$ and $u_1 v_2$ form a *cross* between the $x_1 z$ -path and the $x_2 z$ -path if for $i \in \{1, 2\}$, $u_i \preceq_{x_i z} v_i$ (see Figure 4.2).

Lemma 4.18. *Let x_1, x_2, z be three vertices. If the edges $u_1 v_2$ and $v_1 u_2$ form a cross between the $x_1 z$ -path and the $x_2 z$ -path, then up to symmetry, either $u_1 = v_1$ and $u_2 v_2$ is an edge. Or $u_1 = v_1$ and the $v_2 z$ -path is the edge $v_2 v_1$ concatenated with the $v_1 z$ -path.*

In other words, only cases (c) and (d) of Figure 4.2 can occur.

Proof. For $i \in \{1, 2\}$, we denote by Q_{u_i} (resp. Q_{v_i}) the suffix of the $x_i z$ -path on u_i (resp. v_i). Since suffixes of minimum paths are minimum paths, these four paths are minimum paths. We prove that if a cross does not satisfy the condition of Lemma 4.18, then one of these paths is not minimum.

A *real cross* is a cross for which $u_1 \neq v_1$ and $u_2 \neq v_2$. A *degenerated cross* is a cross for which, up to symmetry, $u_2 = v_2$ and $Q_{v_2} \neq v_2 v_1 Q_{v_1}$.

A real cross satisfies $|Q_{v_1}| = |Q_{v_2}|$. Indeed if $|Q_{v_1}| < |Q_{v_2}|$ then $u_2 v_1 Q_{v_1}$ has length at most $|Q_{v_2}|$. This path is strictly shorter than Q_{u_2} (since Q_{u_2} strictly contains Q_{v_2} while the cross is a real cross), contradicting the minimality of Q_{u_1} .

Consider a degenerated cross for which $u_1 v_1 \notin E$. We have $|Q_{v_2}| \leq |Q_{v_1}|$ otherwise $u_1 v_2 Q_{v_2}$ would be strictly shorter than Q_{u_1} , a contradiction. In addition, $|Q_{v_2}|$ and $|Q_{v_1}|$ differ by at most one since $v_1 v_2$ is an edge. Hence a cross is necessarily of the type Figure 4.2(a), (b), (c) or (d). Remark that (d) is a particular case of (b).

If the cross is of type Figure 4.2(a), then free to exchange x_1 and x_2 , we have $Q_{v_1} \prec_l Q_{v_2}$. So $u_2 v_1 Q_{v_1} \prec_l Q_{u_2}$ (recall that we first compare the last edge) and $|u_2 v_1 Q_{v_1}| \leq |Q_{u_2}|$. Therefore Q_{u_2} is not minimum, a contradiction. So case (a) is cannot occur.

If the cross is of type Figure 4.2(b) and not (d), then we have $Q_{u_2} \prec_l Q_{v_1}$. Otherwise we would have $v_2 v_1 Q_{v_1} \prec_l Q_{v_2}$, and $|u_2 v_1 Q_{v_1}| = |Q_{v_2}|$, and these two paths are distinct since we assume that we

are not in the case of Figure 4.2(d). Hence $u_1 v_2 Q_{v_2} <_l Q_{u_1}$. Finally the type of a cross is necessarily Fig 4.2(c) or (d). \square

Let ℓ be an integer and A, B be two disjoint subsets of vertices. To every pair $(a, b) \in A \times B$, we associate a set of vertices $S_{a,b}$ which is disjoint from $A \cup B$. We say that the family of subsets $\mathcal{S} = \{(S_{a,b})_{(a,b) \in A \times B}\}$ is ℓ -*disconnecting* if for every collection C of \mathcal{S} and every pair (a, b) , we have $d(a, b) > \ell$ in $G \setminus \bigcup C$ if and only if $S_{a,b} \in C$. If such a family of sets exists, then A, B are said to be ℓ -*disconnectable*. Another way of defining ℓ -disconnecting families would be to say that $d(a, b) > \ell$ in $G \setminus S_{a,b}$ and $d(a, b) \leq \ell$ in $G \setminus (\mathcal{S} \setminus S_{a,b})$, or roughly speaking that $S_{a,b}$ is the only set whose deletion can increase $d(a, b)$ above ℓ .

Theorem 4.19. *Let $G = (V, E)$ be a graph and ℓ be an integer. If there exist two subsets A, B of V with $|B| = 2^{|A|}$ which are ℓ -disconnectable, then the VC-dimension of G is at least $|A|$.*

Proof. Let us prove that the set A can be shattered in the B_ℓ -hypergraph of an induced subgraph of G . Associate in a one to one way every vertex b of B to a subset A_b of A . Since A, B are ℓ -disconnectable, there exists a family \mathcal{S} of subsets which is ℓ -disconnecting for A, B . Let C be the collection of \mathcal{S} consisting of all the sets $S_{a,b}$ such that $a \in A_b$. Since \mathcal{S} is ℓ -disconnecting, $B(b, \ell) \cap A = A_b$ in $G \setminus C$, for all $b \in B$. Hence the set A is shattered by balls of radius ℓ in $G \setminus C$. Therefore the VC-dimension of G is at least $|A|$. \square

4.3.2 Sparse sets

Let G be a graph of VC-dimension d and q, ℓ be two integers. Most of the following definitions depends on ℓ . Nevertheless, in order to avoid too heavy notations, this dependence will be implicit on the terminology. A set of balls of radius ℓ is q -*sparse* if no vertex of the graph is in at least q balls of the set. Note that a subset of a q -sparse set is still q -sparse. By abuse of notation, a set X of vertices is called q -*sparse* if the set of balls of radius ℓ centered in X is q -sparse.

Assume that a graph G does not satisfy the $(p, d-1)$ -property for the B_ℓ -hypergraph. Then there exist p balls of radius ℓ such that no vertex is in at least $d-1$ of these p balls, *i.e.* there is a $(d-1)$ -sparse set of size p . In other words, a $(d-1)$ -sparse set of size p is a certificate that the $(p, d-1)$ -property does not hold. In order to prove Theorem 4.15, we just have to show that p can be bounded by a function of d and v_ℓ . In the following, our goal is to prove that the size of a $(d-1)$ -sparse set is at most $f(d, v_\ell)$.

A set X of vertices is d -*localized* if the vertices of X are pairwise at distance at least $\ell+1$ and at most $2\ell - 2^{d+2} - 3$. A pair A, B of disjoint sets of vertices is q -*sparse* if $A \cup B$ is. A disjoint pair A, B of vertices is d -*localized* if the vertices of $A \cup B$ are pairwise at distance at least $\ell+1$, and if for every $a, b \in A \times B$, $d(a, b) \leq 2\ell - 2^{d+2} - 3$. A subpair of a d -localized pair is d -localized. The *size* of a pair A, B is $\min(|A|, |B|)$.

Let us recall Ramsey's theorem.

Theorem 4.20. *(Ramsey) There exists a function r_k such that every complete edge-colored graph G with k colors with no monochromatic clique of size n has at most $r_k(n)$ vertices.*

Lemma 4.21. *Let G be a graph of VC-dimension at most d . There exists a function f such that:*

(a) *Either G contains a d -localized set of size p which is $(d-1)$ -sparse,*

(b) Or the $(f(v_\ell, d, p), d-1)$ -property holds.

Proof. Assume that the $(f(v_\ell, d, p), d-1)$ -property does not hold. Then, there is a $(d-1)$ -sparse set X of size $f(v_\ell, d, p)$. Let $D = 2^{d+2} + 2$. Consider the complete $(D+4)$ -edge-colored graph G' with vertex set X such that, for every $x, y \in X$, xy has color:

- $0 \leq c \leq D$ if $d(x, y) = 2\ell - c$,
- $D+1$ if $d(x, y) \leq \ell$,
- $D+2$ if $d(x, y) > 2\ell$,
- $D+3$ otherwise.

Note that a d -localized set is a monochromatic clique of size $D+3$ (and such a set is $(d-1)$ -sparse since any subset of X is $(d-1)$ -sparse). Let $N = \max(p, v_\ell + 1, 2^{3d+3+\log(4d-2)})$. If $f(v_\ell, d, p) \geq r_{D+4}(N) + 1$, then Theorem 4.20 ensures that there is a monochromatic clique K of size N . Let K' be a clique of color $D+1$ and $x \in K'$. Then $K' \subseteq B(x, \ell) \cap X$. Thus the size of K' is at most $d-2$ since X is $(d-1)$ -sparse. At most v_ℓ balls of radius ℓ centered in X are vertex disjoint by definition of the packing number. Thus the size of a clique of color $D+2$ is at most $v_\ell < N$. The following Claim will ensure that the color c of K does not satisfy $0 \leq c \leq D$.

Claim 4.22. *Let G be a graph and X be a subset of vertices pairwise at distance exactly r . Assume also that no vertex of G belongs to q balls of radius $\lceil r/2 \rceil$ with centers in X . Then the VC-dimension of G is at least $(\log|X| - \log 2q)/3$.*

Proof. Let r' be equal to $\lceil r/2 \rceil$. Free to remove one vertex from X , we can assume that X is even, and we consider a partition A, B of X with $|A| = |B|$. For every pair $(a, b) \in A \times B$, we denote the minimum ab -path by P_{ab} . By abuse of notation, we still denote by G the restriction of G to the vertices of the union of the paths P_{ab} for all $a \in A$ and $b \in B$. Observe that we preserve the hypothesis of Claim 4.22 apart from the fact that the distance between vertices inside A (resp. inside B) may have increased above r . Let y be a vertex of X distinct from a and b . If x belongs to $B(y, r') \cap P_{ab}$, then $d(a, x) \geq \lceil r/2 \rceil$ since $d(a, y) \geq r$ and $d(y, x) \leq \lceil r/2 \rceil$. By symmetry, we also have $d(b, x) \geq \lceil r/2 \rceil$. Hence x is a *midvertex* of P_{ab} , i.e. a vertex of P_{ab} at distance $\lceil r/2 \rceil$ or $\lfloor r/2 \rfloor$ from a (and thus also from b). Recall that a midvertex x of P_{ab} belongs to at most $q-1$ balls of radius r' (including $B(a, r')$ and $B(b, r')$).

Consider the interference matrix $M = (A, B)$ where $m(a, b) = \{y \in (A \cup B) \setminus \{a, b\} \mid B(y, r') \cap P_{ab} \neq \emptyset\}$. Since P_{ab} has at most two midvertices and each of these belongs to at most $q-3$ balls $B(y, r')$ with y different from a and b , the matrix M is a $(2q-6)$ -interference matrix. To avoid tedious calculations and free to increase the interference value, we only assume that M is a $2q$ -interference matrix (with $2q \geq 1$). By Lemma 4.17, there is a proper submatrix M' of size $N = (|X|/2q)^{1/3}$. Let us denote by A' the set of rows and B' the set of columns of the extracted matrix. Let us still denote by G the restriction of the graph to the vertices of the paths $(P_{ab})_{(a,b) \in A' \times B'}$.

Let $a, a' \in A'$ and $b' \in B'$. The key-observation is that if $B(a, r')$ intersects $P_{a'b'}$, then $a = a'$. Indeed assume for contradiction that $x \in B(a, r') \cap P_{a'b'}$. Since $d(a, a') \geq r$, $d(a, b') = r$ and $d(a, x) \leq \lceil r/2 \rceil$, the distance from x to both a' and b' is at least $\lceil r/2 \rceil$, and hence x is a midvertex of $P_{a'b'}$. Thus $a \in m(a', b')$, contradicting the fact that M' is a proper submatrix.

Let M_{ab} be the set of midvertices of P_{ab} , where $a, b \in A' \times B'$. We claim that M_{ab} is disjoint from $P_{a'b'}$, whenever $P_{a'b'} \neq P_{ab}$. Indeed if $x \in M_{ab} \cap P_{a'b'}$, we have in particular both $d(a, x) \leq r'$ and

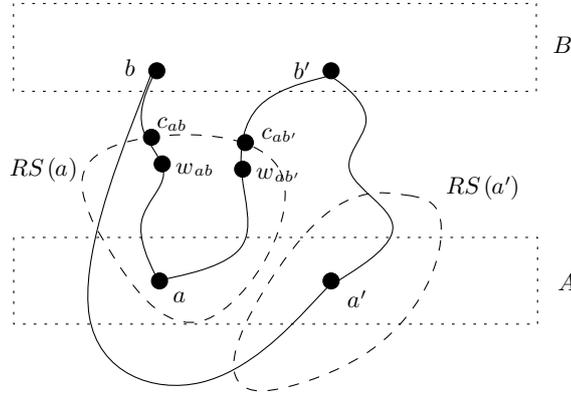


Figure 4.3: Minimum paths with root sections (dashed parts), critical vertices and precritical vertices.

$d(b, x) \leq r'$, and thus by the key-observation $a = a'$ and $b = b'$. In other words, deleting M_{ab} never affects $P_{a'b'}$, whenever $P_{a'b'} \neq P_{ab}$.

Another crucial remark is that every path P of length r from a to b intersects M_{ab} . Indeed, let x be a vertex of P with both $d(a, x) \leq r'$ and $d(b, x) \leq r'$. Since x is in G , it belongs to some path $P_{a'b'}$. By the key-observation, we both have $a' = a$ and $b' = b$, hence $x \in M_{ab}$.

To conclude, observe that deleting M_{ab} increases the distance $d(a, b)$ over r whereas deleting the union of all $M_{a'b'}$ different from M_{ab} does not affect $d(a, b)$ which stays equal to r . Consequently, the sets $(M_{ab})_{(a,b) \in A' \times B'}$ are r -disconnecting for A', B' . Hence, by Theorem 4.19, the VC-dimension of G is at least $\log(N) = (\log|X| - \log 2q)/3$. \square

Since X is $(d-1)$ -sparse, K also is. Then, for every $0 \leq c \leq D$, no vertex of G belongs to $(d-1)$ balls of radius $\lceil (2\ell - c)/2 \rceil \leq \ell$ centered in X . Therefore the color of K cannot be in $0 \leq r \leq D$. Otherwise Claim 4.22 would ensure that the VC-dimension of G is at least $\log(N)/3 - \log(4d - 2)/3 \geq d + 1$. So the clique K of size $N \geq p$ has color $D + 3$. \square

In d -localized set, the vertices have to be pairwise at distance at least $d + 1$ and at most $2\ell - 2^{d+2} - 3$. The edge colored graph of Lemma 4.21 was constructed in order to ensure this property. A set is (α, β) -localized if the vertices are pairwise at distance at least $\ell + \alpha$ and at most $2\ell - \beta$. A slight modification of the previous proof ensures that the same holds for (α, β) -localized sets. Note nevertheless that the function f will depend on v_ℓ, α, β and p .

4.3.3 Localized and independent pairs

In this section we introduce a notion of independence for every pair of vertices. We first give some properties of independent pairs and we will finally show that any large enough $(d-1)$ -sparse and localized pair contains an independent subpair large enough.

Let A, B be a d -localized pair. In the following we consider ab -paths with $a \in A$ and $b \in B$. Recall that the ab -path is the minimum path with minimum lexicographic order from a to b . Note that $a \in A$ is not a vertex of the $a'b'$ -path if $a' \neq a$. Indeed the $a'b'$ -path has length at most $2\ell - 7$ since

A, B is d -localized. And $d(a, a') > \ell$ and $d(a, b') > \ell$, so a $a'b'$ -path passing through a would have length at least $2\ell + 2$.

For every pair $a, b \in A \times B$, the *critical vertex* c_{ab} (resp. c_{ba}) is the vertex of the ab -path at distance $\ell - 3$ from a (resp. b) and the *precritical vertex* w_{ab} is the vertex of the ab -path at distance $\ell - 4$ from a (see Figure 4.3). Such vertices exist since $d(a, b) > \ell$ (and is unique by minimality of the ab -path). Note that c_{ab} and w_{ab} are adjacent. The *root section of* $a \in A$ (resp. $b \in B$), denoted by $RS(a)$ (resp. $RS(b)$), is the set of vertices of the ac_{ab} -subpaths (resp. $c_{ba}b$ -subpaths) of the ab -paths for all $b \in B$ (resp. $a \in A$). Note that these notions are asymmetrical since we only consider minimum AB -paths and not minimum BA -paths. We denote by $RS(A)$ the set $\bigcup_{a \in A} RS(a)$.

Observation 4.23. *Let A, B be a d -localized pair. For every a, b in $A \times B$, the critical vertex c_{ab} and the precritical vertex w_{ab} are in $RS(b)$.*

Proof. Since the ab -path is minimum, $d(b, c_{ab}) = d(a, b) - d(a, c_{ab})$. Since A, B is d -localized, $d(a, b) \leq 2\ell - 7$. So $d(b, c_{ab}) \leq 2\ell - 7 - (\ell - 3) \leq \ell - 4$. And $d(b, w_{ab}) \leq \ell - 3$ since $w_{ab}c_{ab}$ is an edge. Thus both c_{ab} and w_{ab} are in $RS(b)$. \square

All the vertices of the AB -paths are in a root section. Indeed the vertices of the prefix path on c_{ab} of the ab -path are in $RS(a)$ by definition. The others are in $RS(b)$ since c_{ab} is in $RS(b)$.

A d -localized pair A, B is *independent*, if for every $a, b \in A \times B$, $B(c_{ab}, \ell) \cap (A \cup B) = \{a, b\}$ and $B(c_{ba}, \ell) \cap (A \cup B) = \{a, b\}$. A subpair of an independent pair is still independent. In addition, A, B is still independent in the graph restricted to the AB -paths. Before proving that a d -localized pair contains large independent subpairs (Lemma 4.26), let us state general properties on independent pairs.

Lemma 4.24. *Let A, B be an independent pair.*

(a) *Two endpoints disjoint AB -paths are at distance at least 4.*

(b) *For every pair a, a' in A (resp. b, b' in B), $d(RS(a), RS(a')) \geq 4$ (resp. $d(RS(b), RS(b')) \geq 4$).*

Proof. Let us first prove (b). We prove it for vertices of A , the case of vertices of B is handle symmetrically. Let $a \neq a'$ with $u \in RS(a)$ and $v \in RS(a')$. There exists b and b' in B such that u is in the prefix path on c_{ab} of the ab -path and v is in the prefix path on $c_{a'b'}$ of the $a'b'$ -path. Free to exchange a and a' , $d(a, u) \leq d(a', v)$. Since $d(a', c_{a'b'}) = \ell - 3$, we have $d(a, u) + d(v, c_{a'b'}) \leq d(a', v) + d(v, c_{a'b'}) = \ell - 3$. Since A, B is independent, $d(a, c_{a'b'}) > \ell$ thus $\ell < d(a, u) + d(u, v) + d(v, c_{a'b'}) \leq \ell - 3 + d(u, v)$, i.e. $d(u, v) \geq 4$. So (b) holds.

Let u be a vertex of the ab -path, and v be a vertex of the $a'b'$ -path such that $a \neq a'$ and $b \neq b'$. Lemma 4.24(b) ensures that, up to symmetry, $u \in RS(a)$ and $v \in RS(b')$. In addition, we can assume that $d(a, u) \leq d(b', v)$. So $d(a, u) + d(v, c_{b'a'}) \leq d(b', v) + d(v, c_{b'a'}) = \ell - 3$. So $\ell < d(a, c_{a'b'}) \leq d(a, u) + d(u, v) + d(v, c_{b'a'}) \leq \ell - 3 + d(u, v)$. Hence $d(u, v) \geq 4$. \square

An edge *leaves* a set S if one of its endpoints is in S and not the other one.

Observation 4.25. *Let A, B be an independent pair and $a \in A$. For all $b \neq b'$, we have $w_{ab} \neq w_{ab'}$ (and then $c_{ab} \neq c_{ab'}$). Furthermore the edges of the aB -paths leaving $RS(a)$ form an induced matching.*

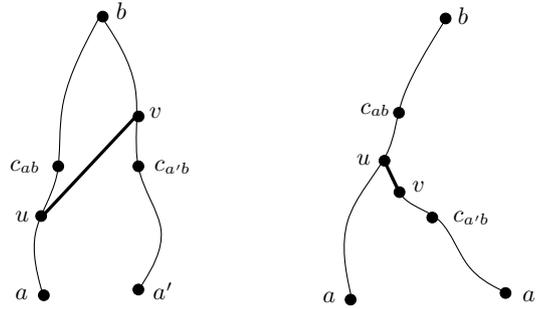


Figure 4.4: Examples of escapes. The right one is an edge of the $a'b$ -path.

Proof. Observation 4.23 ensures that $w_{ab} \in RS(b)$ and $w_{ab'} \in RS(b')$. So Lemma 4.24(b) ensures that $w_{ab} \neq w_{ab'}$. When two minimum paths from the same vertex separate, they do not meet again. Hence the edges of aB -paths leaving $RS(a)$ are vertex disjoint, *i.e.* they form a (non necessarily induced) matching. By Observation 4.23, the edge of the ab -path leaving $RS(a)$ is an edge with both endpoints in $RS(b)$. Thus Lemma 4.24(b) ensures that the matching is induced. \square

Lemma 4.26. *Let G be a graph. The size of a $(d-1)$ -sparse and d -localized pair with no independent subpair of size p is at most $(2d-4)p^3$.*

Proof. Let A, B be a $(d-1)$ -sparse and d -localized pair of size $(2d-4)p^3 + 1$. For every vertex u , $I(u)$ denotes $B(u, \ell) \cap (A \cup B)$. Since $A \cap B = \emptyset$ and since both c_{ab} and c_{ba} exist, the matrix $M = (A, B)$, where $m(a, b) = (I(c_{ab}) \cup I(c_{ba})) \setminus \{a, b\}$, is a well-defined interference matrix. The pair A, B is $(d-1)$ -sparse, then $|I(u)| \leq d-2$. Thus M is a $(2d-4)$ -interference matrix.

By Lemma 4.17, M has a proper submatrix (A', B') of size p . Thus for every $a', b' \in A' \times B'$, $B(c_{a'b'}, \ell) \cap (A' \cup B') = \{a', b'\}$ and the same holds for $c_{b'a'}$, *i.e.* A', B' is independent. \square

4.3.4 Escape property

Let A, B be an independent pair. An edge uv between two AB -paths is an *escape* from a if uv leaves $RS(a)$ and uv is not an edge of the ab -path for some $b \in B$ (see Figure 4.4). By convention, when uv is an escape from a , v still denotes the vertex which is not in $RS(a)$. Remark that an escape can be an edge of a minimum path (see Figure 4.4).

Let uv be an escape from a . The vertex v is on some $a'b'$ -path. Lemma 4.24(b) ensures that $a = a'$ or $b = b'$. If $a = a'$ then $d(a, u) = \ell - 3$. Indeed otherwise v is in some ab' -path and $v \notin RS(a')$, *i.e.* $d(a, v) \geq \ell - 2$. The minimality of the ab' -path ensures that $d(u, a) = \ell - 3$, *i.e.* u is the vertex c_{ab} . Though Observation 4.25 ensures that there is no edge between c_{ab} and v . So $a \neq a'$, *i.e.* $b = b'$. Thus uv is called an *escape from a to a' for b* . On Figure 4.4, both edges uv are escapes from a to a' for b .

If a vertex x has a neighbor in $RS(a)$, a is called an *origin section* on x . Lemma 4.24(b) ensures that every vertex has at most one origin section. Note that if uv is a escapes from a , then a is the origin section of v .

A *deep escape* is an escape such that u is neither a critical vertex nor a precritical vertex. The *(deep) escape graph of b* is a directed graph with vertex set A and with an arc aa' if there is a (deep) escape from a to a' for b .

Lemma 4.27. *Let A, B be an independent pair. For every $b \in B$, the escape graph of b has no circuit.*

Proof. Assume that there is a circuit $a_0, a_1, \dots, a_k, a_0$. In the following indices have to be understood modulo $k + 1$. For every i , let $u_i v_i$ be an escape from a_i to a_{i+1} for b . Since $u_i \in RS(a_i)$ and $u_{i+1} \in RS(a_{i+1})$, Lemma 4.24(b) ensures that $d(u_i, u_{i+1}) \geq 4$, then $d(v_i, u_{i+1}) \geq 3$. Hence $d(b, u_i) \leq d(b, v_i) + 1 < d(b, v_i) + d(v_i, u_{i+1}) = d(b, u_{i+1})$. A propagation of these inequalities along the arcs of the circuit leads to $d(b, u_0) < d(b, u_0)$. \square

The deep escape graph of b is a subgraph, in sense of arcs, of the escape graph of b . Thus Lemma 4.27 ensures that the deep escape graph of b is also acyclic. For every b , an *order inherited from b* is a partial order on A such that if there is an escape from a to a' for b then $a < a'$. When the escape graph of b is a transitive tournament, such an order is unique. An independent pair A, B has the *escape property* if for every $b \in B$, the deep escape graph of b is a transitive tournament.

Let $u_{a'b} v_{a'b}$ be a *first-in escape to a' for b* , i.e. an escape to a' for b satisfying $v_{a'b} \trianglelefteq_{a'b} v$ for every escape uv to a' for b . In other words it is the first edge of the $a'b$ -path which is in-escape to a' . The vertex $v_{a'b}$ is called the *incoming vertex* of the $a'b$ -path. Note that several first-in escapes to a' can exist, but the incoming vertex is unique. The *free section* of the $a'b$ -path, denoted by $FS(a', b)$, is the $c_{a'b} v_{a'b}$ -subpath of the $a'b$ -path where $c_{a'b}$ is not included but $v_{a'b}$ is included. Lemma 4.24(b) ensures the free section exists.

Lemma 4.28. *Let A, B be a pair satisfying the escape property. Then there is no edge between two free sections of AB -paths.*

Proof. Let us prove that such an edge would provide a forbidden cross. Consider an edge xy such that $x \in FS(a', b')$ and $y \in FS(a, b)$. Recall that $x \in RS(b')$ and $y \in RS(b)$. So Lemma 4.24(a) ensures that $b = b'$ and y is on the $a'b$ -path for $a' \neq a$. Assume w.l.o.g. that a is less than a' in the total order on A given by the escape property. So there is a deep escape uv from a to a' for b . Since $y \triangleright_{ab} c_{ab}$ and u is distinct from c_{ab}, w_{ab} (by definition of deep escape), we have $d(u, y) \geq 3$ by Lemma 4.24(b). And $x \trianglelefteq_{a'b} v$. So edges xy and uv contradicts Lemma 4.18. \square

Note that we need a deep escape in order to have a forbidden cross. Indeed, if the escape is not deep, then the resulting cross could not be forbidden. It is the unique reason we need deep escapes. In the following, we do not use anymore that there exist deep escapes.

Lemma 4.29. *Let G be a graph of VC-dimension at most d . The size of an independent pair with no subpair of size 2^{d+1} satisfying the escape property is at most $r_{2^{d+2}}(2^{d+1})$.*

Proof. Let (A, B) be a pair of size $r_{2^{d+2}}(2^{d+1}) + 1$.

Claim 4.30. *A, B has a subpair X, Z of size 2^{d+1} such that:*

- (1) *either X, Z does not contain a deep escape,*
- (2) *or X, Z satisfies the escape property.*

Proof. Let $B' = \{b_1, \dots, b_{2^{d+2}}\}$ be a subset of B of size 2^{d+2} . Consider the complete edge-colored graph G' on vertex set A . The colors are binary integers of 2^{d+2} digits. The i -th digit of the color of aa' is 1 if there is a deep escape from a to a' (or from a' to a) for b_i and 0 otherwise. Theorem 4.20 ensures that G' contains a monochromatic clique X of size 2^{d+1} . Let us denote by c the color of the edges of $G'[X]$. A subset \mathcal{I} of 2^{d+1} digits of c are equal. Let $Z = \{b_i, i \in \mathcal{I}\}$. If $c(i) = 0$ for every $i \in \mathcal{I}$ then (1) holds for the pair X, Z , otherwise (2) holds. \square

Let us prove that Claim 4.30(1) cannot hold. Let X, Z be an independent pair with no deep escape. Consider the restriction of the graph to the vertices of the xz -paths for $x, z \in X \times Z$. For every x, z , we denote by $CS(x, z)$ the set of vertices exclusively on the xz -path.

Claim 4.31. $CS(x, z)$ separates x from c_{xz} and from w_{xz} in the graph induced by $RS(x)$.

Proof. Let us first prove it for c_{xz} . We can assume that $c_{xz} \notin CS(x, z)$. Thus there exist x', z' such that the $x'z'$ -path passes through c_{xz} . By Lemma 4.24(a), $x = x'$ or $z = z'$. If $x = x'$ then $c_{xz} = c_{xz'}$, contradicting Observation 4.25. Hence $z = z'$. Let P be a path from x to c_{xz} and let u be the last vertex of P which is not on some minimum $x''z$ -path for $x'' \neq x$. The vertex u exists since c_{xz} is on the $x'z$ -path and x is not on such a path as underlined in Section 4.3.3. The vertex u is in $CS(x, z)$. Indeed, otherwise the xz' -path passes through u for some $z \neq z'$. And by definition of u , u has a neighbor in some $x'z$ -path which contradicts Lemma 4.24(a).

Let P be a path from x to w_{xz} which does not pass through $CS(x, z)$. Since $w_{xz} \notin CS(x, z)$, some $x'z'$ -path passes through w_{xz} . By Lemma 4.24(a), $x = x'$ or $z = z'$. By Observation 4.25, we have $z = z'$. Then some $x'z$ -path passes through w_{xz} and then through c_{xz} since two minimum path to z which intersect on u coincide between u and z . So $c_{xz} \notin CS(x, z)$. Hence P plus the edge $w_{xz}c_{xz}$ is a path from x to c_{xz} which does not pass through $CS(x, z)$. \square

Claim 4.32. The VC-dimension of a graph with an independent pair of size d is at least $\log d$.

Proof. Let X, Z be an independent pair of size d . To prove it, we prove that X, Z is $(2\ell - 5)$ -disconnectable with the sets $CS(x, z)$. Let $x \in X$. Since there is no deep escapes, an edge leaving $RS(x)$ intersect a critical or a precritical vertex. Since X, Z is independent, if $z \neq z'$ then $d(c_{xz'}, z) \geq \ell + 1$, so $d(w_{xz'}, z) \geq \ell$. By definition $d(x, c_{xz'}) = \ell - 3$ and $d(x, w_{xz'}) = \ell - 4$. Thus, for every $z' \neq z$, the length a path from x to z passing through $c_{xz'}$ or $w_{xz'}$ is at least $2\ell - 4$. Hence every path of length at most $2\ell - 5$ from x to z passes through c_{xz} or w_{xz} .

Thus, by Claim 4.31, there is no path of length at most $2\ell - 5$ from x to z in $G(V \setminus CS(x, z))$. By definition, the sets $CS(x, z)$ are disjoint and do not intersect the $x'z'$ -path if $x \neq x'$ or if $z \neq z'$. Finally, since X, Z is d -localized, $d(x, z) \leq 2\ell - 5$. Hence the sets $CS(x, z)$ are $(2\ell - 5)$ -disconnecting for X, Z . By Theorem 4.19, the VC-dimension is at least $\log d$. \square

A subpair of an independent pair is independent. Hence Claim 4.32 ensures that Claim 4.30(1) cannot hold. Otherwise the VC-dimension would be at least $d + 1$. So Claim 4.30(2) holds. \square

Almost all statements of Section 4.3.3 and 4.3.4 asked for a d -localized pair. Nevertheless the proofs are still correct with a weaker definition of d -localized pair. A pair is *weakly localized* if vertices are pairwise at distance at most $2\ell - 7$ instead of $2\ell - 2^{d+2} - 3$. Note that since $d \geq 0$, a d -localized pair is weakly localized. The stronger condition of d -localized sets on the lengths of the paths will be used in Section 4.3.5.

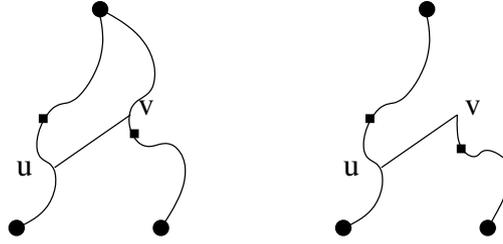


Figure 4.5: The minimum path (at the left) is transformed into the jump path (at the right).

4.3.5 Escape property implies large VC-dimension

The outline of the proof of Lemma 4.29 consists in finding a $(2\ell - 5)$ -disconnecting pair. The approach will be the same when the escape property holds even if the proof will be a little bit more technical since we first need to re-root some paths.

Theorem 4.33. *Let G be a graph of VC-dimension d . The size of a pair with the escape property is at most $2^{d+1} - 1$.*

Proof. Assume by contradiction that a pair A, B of size 2^{d+1} satisfies the escape property. Let $b \in B$. Order the vertices of A along the order inherited from b . We denote by $u_i v_i$ the first-in escape to a_i for b . Recall that by convention, v_i denotes the incoming vertex of the $a_i b$ -path.

By definition of order inherited from b , there exists $j < i$ such that $u_i \in RS(a_j)$. So u_i is on prefix path on c_{ij} (and then on a_j since $c_{a_j b} \triangleright_{a_j b} a_j$) of the $a_j b$ -path. Therefore the following new collection of Ab -paths, called jump paths (for b), are well-defined. This collection is constructed by induction. The *jump path* of $a_1 b$ is the $a_1 b$ -path. The *jump path* of $a_i b$ is the prefix path on v_i of the $a_i b$ -path, the edge $v_i u_i$ of origin section a_j and the suffix path on u_i of the jump path of $a_j b$ (see Figure 4.5). Jump paths can be equal to minimum paths (see the right of Figure 4.4).

Note that the jump paths follow the edges of minimum AB -paths except on incoming vertices. An edge which is in the jump path of ab but not on some minimum $a'b$ -path is called a *reroot*. Every jump path is rerooted at most $|A| = 2^{d+1}$ times since a jump path is re-rooted on a jump path for the same b of strictly smaller index. In addition each re-root increases the length of the path by at most two since $|d(u_i, b) - d(v_i, b)| \leq 1$ ($u_i v_i$ is an edge). Since A, B is d -localized, the length of the jump path of ab is at most $(2\ell - 2^{d+2} - 3) + 2^{d+1} \cdot 2 = 2\ell - 3$.

Claim 4.34. *Jump paths only contain vertices of $RS(A)$ and of free sections. In addition jump paths for b only contain vertices of ab -paths for $a \in A$.*

Proof. By induction on the order inherited from b . It holds for the jump path of $a_1 b$. The jump path of $a_i b$ coincides with the $a_i b$ -path from a to the incoming vertex, *i.e.* on $RS(a_i)$ and on $FS(a_i, b)$. And by induction, it holds for the remaining vertices. \square

Consider the restriction of the graph to the vertices of the jump paths for every $a, b \in A \times B$.

Claim 4.35. *An edge leaving $FS(a_i, b)$ has an endpoint in $RS(a_j) \cup c_{a_i b}$, where a_j denotes the origin section of the incoming vertex v_i .*

Proof. Claim 4.34 ensures that every vertex is either in $RS(A)$ or in $FS(a, b)$ for some a, b . By Lemma 4.28, there is no edge between two free sections. So an edge leaving $FS(a_i, b)$ has an endpoint in $RS(x)$. Observation 4.25 ensures that an edge with an endpoint in $FS(a_i, b)$ and the other in $RS(a_i)$ has endpoint $c_{a_i, b}$ (the edges of minimum paths from a leaving $RS(a)$ induces a matching). Since there is no escape to a_i before the incoming vertex, there are only edges to the root section of the origin vertex. \square

Claim 4.36. *The vertex c_{ab} is in every path from a to b of length at most $2\ell - 3$.*

Proof. Let us first prove the following remark. Let P be a path to b passing through $RS(a)$ such that:

- P cannot leave definitively $RS(a)$ through $c_{ab'}$ for any $b' \in B$ nor through $FS(a, b)$.
- P cannot pass through $c_{a'b'}$ if both a', b' are distinct from a, b .

Then P passes through $FS(a', b)$ and then $RS(a')$. Since P cannot leave $RS(a)$ through a critical vertex, it must leave it through an escape. The other endpoint of the escape must be in a free section by Claim 4.34. The path P cannot leave definitively $RS(a)$ through an escape for $b' \neq b$. Otherwise by Claim 4.35, it would leave $FS(a', b')$ through $c_{a'b'}$ for some $b' \neq b$, a contradiction. So it leaves it through a shortcut from a , *i.e.* it passes through $FS(a', b)$. Finally Claim 4.35 ensures that it passes through $FS(a', b)$ and enter in $RS(a')$.

Consider a path P from a to b of length at most $2\ell - 3$. The path P cannot contain a vertex $c_{a'b'}$ for $b' \neq b$. Otherwise P would have length at least $2\ell - 2$ since $d(c_{a'b'}, a) \geq \ell - 3$ and $d(b, c_{a'b'}) \geq \ell + 1$. Assume in addition that P does not contain c_{ab} . Since the unique neighbor of $FS(a, b)$ in $RS(a)$ is c_{ab} , P cannot definitively leave $RS(a)$ through $FS(a, b)$. Therefore, the remark ensures that P passes through $RS(a', b)$ and $RS(a')$ for some $a' \neq a$. Since the vertices of $FS(a', b)$ already are in P , P cannot definitively leave $RS(a')$ through $FS(a', b)$. So, by induction, P cannot leave definitively $RS(A)$. Indeed every time P leaves definitively $RS(a')$, the remark ensures that P enters in some $RS(a'')$ for some $a'' \in A$. So c_{ab} is in P . \square

Let $CS(a, b)$ be the set of vertices which are only on the jump path of ab .

Claim 4.37. *All the paths of length at most $2\ell - 3$ from a to b pass through $CS(a, b)$.*

Proof. By Claim 4.36, every path from a to b passes through c_{ab} . If $c_{ab} \in CS(a, b)$, then Claim 4.37 holds. So the jump path of $a'b'$ passes through c_{ab} . Jump paths coincide with minimum paths until their incoming vertices. So $a \neq a'$ since $c_{ab} \neq c_{ab'}$. In addition $b = b'$ otherwise Lemma 4.24(b) would ensure that $d(a', c_{ab}) > \ell$ and $d(b', c_{ab}) > \ell$, a contradiction since the length of the jump path of $a'b'$ is at most $2\ell - 3$. So c_{ab} is on the jump path of $a'b$.

Let P be a path from a to c_{ab} which does not pass through $CS(a, b)$. Let u be the last vertex which is not in a jump path of $a'b$ for some $a' \neq a$, and let v be its neighbor. The vertex u exists since c_{ab} is in a jump path of $a'b$ and not a . Let us prove by contradiction that $u \in CS(a, b)$. Assume that the jump path of $a'b'$ passes through u for some $b' \neq b$. Assume that $a' \neq a$. Then $d(a', u) > \ell$ since $u \in RS(a)$. And $d(b', u) > \ell$ since $v \in RS(b)$. So we can assume that $a' = a$.

There exists $a' \neq a$ in A and b in B such that v is on the $c_{a'b'}b$ -subpath of the jump path of $a'b$. But since $u \in RS(a)$, we have $v \notin RS(a')$ then u is on the $ac_{ab'}$ -subpath of the jump path of ab' . The two following inequalities, illustrated on Figure 4.6, provide a contradiction.

First $d(v, c_{a'b}) < d(u, c_{ab'}) + 1$ since $d(b, c_{a'b}) \leq \ell$ and $d(b, c_{ab'}) > \ell$ and uv is an edge. Indeed the first inequality provide from the fact that the jump path have length at most $2\ell - 3$ and the second

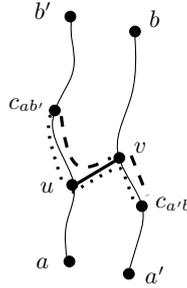


Figure 4.6: Illustration of Claim 4.37. The dotted and dashed paths represents the two inequalities.

from the independence of A, B .

Second $d(u, c_{ab'}) + 3 < d(v, c_{a'b}) + 1$ since $d(a, c_{ab'}) \leq \ell - 3$ and $d(a, c_{a'b}) > \ell$. Indeed the first inequality provides from the fact that jump path are not modified before critical vertices and the second from the independence.

So $d(v, c_{a'b}) - 1 < d(u, c_{ab'}) < d(v, c_{a'b}) - 2$ which is impossible. \square

To conclude we apply Theorem 4.19 with the sets $CS(x, z)$ for paths of length $2\ell - 3$. Indeed, by definition the sets $CS(x, z)$ are pairwise disjoint and are only on the jump path of xz . Claims 4.36 and 4.37 ensure that the sets $CS(x, z)$ are $(2\ell - 3)$ -disconnecting for X, Z . Therefore the graph G must have VC-dimension at least $d + 1$ which is impossible. \square

4.4 Conclusion

We prove that several classes of graphs have a bounded VC-dimension, *i.e.* are simple for the iterated neighborhood point of view. It could be interesting to understand which classes of graphs are simple. Indeed, a bounded VC-dimension ensures some structure on the graph. In particular it means that the neighborhoods cannot intersect in every way and that the graph is well-structured.

Let us finally end this chapter with several open problems. In graph coloring, we also need some structure in order to ensure that there is some bounds on the chromatic number. Dvůrák and Král proved in [78] that graphs of bounded rankwidth are χ -bounded. Actually they prove it for classes of graphs with cuts of small rank, *i.e.* for graphs which have some structure of the cuts. The same might be extended for graphs of bounded VC-dimension.

Conjecture 7. *If a class of graphs has its VC-dimension bounded by a function of its maximal clique then the class is χ -bounded.*

A weaker version of this conjecture consists in proving it for graphs of bounded 2VC-dimension. Note that the reverse of this conjecture does not hold. Indeed, perfect graphs do not have a bounded VC-dimension. Indeed consider a clique K_n on n vertices. For every subset of K_n , create a new vertex connected exclusively to this subset. One can easily verify that K_n is shattered in the B_1 -hypergraph. In addition, one can easily check that this graph contains neither an odd hole nor an odd antihole.

Note that the complexity of the previous construction is hidden by the clique. Hence, it could be interesting to study triangle-free classes from the VC-dimension point of view. We conjecture that the following classes, known to be χ -bounded, have a bounded VC-dimension.

Conjecture 8. *The class of graphs with no induced P_ℓ has a VC-dimension bounded by ℓ and the size of a maximum clique.*

The circle graphs with a bounded maximum clique has a bounded VC-dimension.

Separate cliques and stable sets

This Chapter is the presentation of two joint works with Aurélie Lagoutte and Stéphan Thomassé (see [37, 38]).

The main part of this chapter consists in studying a Yannakakis' conjecture on the number of separators needed for separating cliques and stable sets in graphs. We prove that this conjecture is equivalent with another conjecture of Alon, Saks and Seymour (one implication was already known). We also link these two conjectures with conjectures on constraint satisfaction problems. Finally we prove that the Yannakakis' conjecture holds for random graphs, split-free graphs. The proof for split-free graphs is based on a VC-dimensional argument (Section 5.2.2). Recall that a F -free graph is a graph which does not contain any induced copy of F .

In the second one, we prove that the Erdős-Hajnal conjecture holds for $(P_k, \overline{P_k})$ -free graphs (the proof is presented in Section 5.2.3). This result implies that the Yannakakis' conjecture holds for $(P_k, \overline{P_k})$ -induced free graphs.

All along this chapter we will use the term “cut” instead of separator. We have seen that a separator is a bipartition of the vertex set in which there is an asymmetry: indeed we favor a vertex called the root which is always on the separator. In the following we do not want to favor vertices or to find separators of minimum borders as we did in Chapter 1 and 2. So, in order to avoid confusions, a bipartition of the vertex set will be called a *cut* in this chapter.

cut

5.1 Introduction

The goal of this Chapter is twofold. First, we focus on the Clique-Stable Set separation problem and provide classes of graphs for which polynomial separators exist. Then we show that this classical problem from communication complexity is equivalent to one in graph theory and one in CSP. Let us make a brief overview of each domain focusing on the problem.

Communication complexity and the Clique-Stable Set separation. Yannakakis introduced in [191] the following communication complexity problem, called *Clique versus Independent Set* (CL-

CL-IS

IS for brevity): given a publicly known graph Γ on n vertices, Alice and Bob agree on a protocol, then Alice is given a clique and Bob is given a stable set. They do not know which clique or which stable set was given to the other one, and their goal is to decide whether the clique and the stable set intersect or not, by minimizing the worst-case number of exchanged bits. Recall that the intersection of a clique and a stable set is at most one vertex. In the deterministic version, Alice and Bob alternatively send messages to each other, and the minimization is on the number of bits exchanged between them. It is a long standing open problem to prove a $\mathcal{O}(\log^2 n)$ lower bound for the deterministic communication complexity. In the non-deterministic version, a prover knowing the clique and the stable set sends a certificate in order to convince both Alice and Bob of the right answer. Then, Alice and Bob exchange one final bit, saying whether they agree or disagree with the certificate. The aim is to minimize the size of the certificate.

In this particular setting, a certificate proving that the clique and the stable set intersect is just the name of the vertex in the intersection. Such a certificate clearly has logarithmic size. Convincing Alice and Bob that the clique and the stable set do not intersect is much more complicated. A certificate can be a bipartition of the vertices such that the whole clique is included in the first part, and the whole stable set is included in the other part. Such a partition is a cut that separates the clique and the stable set. A family \mathcal{F} of m cuts such that for every disjoint clique and stable set, there is a cut in \mathcal{F} that separates the clique and the stable set is called a CS-separator of size m . Observe that Alice and Bob can agree on a CS-separator at the beginning, and then the prover just gives the name of a cut that separates the clique and the stable set: the certificate has size $\log_2 m$. Hence if there is a CS-separator of polynomial size in n , one can ensure a non-deterministic certificate of size $\mathcal{O}(\log_2 n)$.

Yannakakis proved that there is a $c \log_2 n$ certificate for the *CL-IS* problem if and only if there is a CS-separator of size n^c . The existence of such a CS-separator is called in the following the Clique-Stable Set separation problem. The best upper bound so far, due to Hajnal (cited in [138]), is the existence for every graph G of a CS-separator of size $n^{(\log n)/2}$. The *CL-IS* problem arises from an optimization question which was studied both by Yannakakis [191] and by Lovász [139]. The question is to determine if the stable set polytope of a graph is the projection of a polytope in higher dimension, with a polynomial number of facets (called extended formulation). The existence of such a polytope in higher dimension implies the existence of a polynomial CS-separator for the graph. Moreover, Yannakakis proved that the answer is positive for several subclasses of perfect graphs, such as comparability graphs and their complements, chordal graphs and their complements, and Lovász proved it for a generalization of series-parallel graphs called t -perfect graphs. The existence of an extended formulation for general graphs has recently been disproved by Fiorini et al. [91], and is still open on perfect graphs.

Graph coloring and the Alon-Saks-Seymour conjecture. Given a graph G , the bipartite packing, denoted by \mathbf{bp} , is the minimum number of edge-disjoint complete bipartite graphs needed to partition the edges of G . The Alon-Saks-Seymour conjecture (cited in [122]) states that if a graph has bipartite packing k , then its chromatic number χ is at most $k+1$. It is inspired from the Graham-Pollak theorem [107] which states that $\mathbf{bp}(K_n) = n - 1$. Huang and Sudakov proposed in [120] a counterexample to the Alon-Saks-Seymour conjecture (then generalized in [60]), twenty-five years after its statement. Actually they proved that there is an infinite family of graphs for which $\chi \geq \mathbf{bp}^{6/5}$. Amano

$$\begin{pmatrix} 0 & * & 0 & * \\ * & 0 & * & * \\ 0 & * & * & * \\ * & * & * & 1 \end{pmatrix}$$

Figure 5.1: Matrix M for the stubborn problem.

improved this result in [12] by proving that some graphs have chromatic number at least $\mathbf{bp}^{3/2}$. The Alon-Saks-Seymour conjecture can now be restated as the *polynomial* Alon-Saks-Seymour conjecture: is the chromatic number polynomially upper bounded in terms of \mathbf{bp} ? Moreover, Alon and Haviv [7] observed that a gap $\chi \geq \mathbf{bp}^c$ for some graphs would imply a n^c lower bound for the Clique-Stable Set separation problem. Consequently, Huang and Sudakov's result gives a $n^{6/5}$ lower bound. This in turn implies a $6/5 \log_2(n) - \mathcal{O}(1)$ lower bound on the non-deterministic communication complexity of $CL - IS$ when the clique and the stable set do not intersect.

A generalization of the bipartite packing of a graph is the t -biclique number, denoted by \mathbf{bp}_t . It is the minimum number of complete bipartite graphs needed to cover the edges of the graph such that each edge is covered at least once and at most t times. It was introduced by Alon [5] to model neighborly families of boxes, and the most studied question so far is finding tight bounds for $\mathbf{bp}_t(K_n)$.

Constraint satisfaction problem and the stubborn problem. The complexity of the so-called *list- M partition problem* has been widely studied in the last decades (see [179] for an overview). M stands for a fixed $k \times k$ symmetric matrix filled with 0, 1 and *. The input is a graph $G = (V, E)$ together with a list assignment $\mathcal{L} : V \rightarrow \mathcal{P}(\{A_1, \dots, A_k\})$ and the question is to determine whether the vertices of G can be partitioned into k sets A_1, \dots, A_k respecting two types of requirements. The first one is given by the list assignments, that is to say v can be put in A_i only if $A_i \in \mathcal{L}(v)$. The second one is described in M , namely: if $M_{i,i} = 0$ (resp. $M_{i,i} = 1$), then A_i is a stable set (resp. a clique), and if $M_{i,j} = 0$ (resp. $M_{i,j} = 1$), then A_i and A_j are completely non-adjacent (resp. completely adjacent). If $M_{i,i} = *$ (resp. $M_{i,j} = *$), then A_i can be any set (resp. A_i and A_j can have any kind of adjacency).

Feder et al. [88, 89] proved a *quasi-dichotomy theorem*. The list- M partition problems are classified between NP-complete and quasi-polynomial time solvable (*i.e.* time $\mathcal{O}(n^{c \log n})$ where c is a constant). Moreover, many investigations have been made about small matrices M ($k \leq 4$) to get a *dichotomy theorem*, meaning a classification of the list- M partition problems between polynomial time solvable and NP-complete. Cameron et al. [44] reached such a dichotomy for $k \leq 4$, except for one special case (and its complement) then called the *stubborn problem* (the corresponding symmetric matrix has size 4; $M_{1,1} = M_{2,2} = M_{1,3} = M_{3,1} = 0$, $M_{4,4} = 1$; the other entries are *), which remained only quasi-polynomial time solvable. Cygan et al. [65] closed the question by finding a polynomial time algorithm solving the stubborn problem. More precisely, they found a polynomial time algorithm for 3-COMPATIBLE COLORING, which was introduced in [87] and said to be no easier than the stubborn problem. 3-COMPATIBLE COLORING has also been introduced and studied in [129] under the name ADAPTED LIST COLORING, and was proved to be a model for some strong scheduling problems. It is defined in the following way:

3-COMPATIBLE COLORING PROBLEM (3-CCP)

Input: An edge coloring f_E of the complete graph on n vertices with 3 colors $\{A, B, C\}$.

Question: Is there a coloring of the vertices with $\{A, B, C\}$, such that no edge has the same color as both its endpoints?

Contribution of this Chapter. All along this Chapter, the Clique-Stable Set separation problem will be considered as our reference problem. More precisely, we start in Section 5.2 by proving that there is a polynomial CS-separator for three classes of graphs: random graphs, induced split-free graphs and graphs with no induced path of length k nor its complement. The proof for random graphs is based on random cuts. In the second case, it is based on Vapnik-Chervonenkis dimension. In the last one, it will be a consequence of the proof of the Erdős-Hajnal conjecture for graphs with no induced path of length k nor its complement.

In Section 5.3, we extend Alon and Haviv's observation and prove the equivalence between the polynomial Alon-Saks-Seymour conjecture and the Clique-Stable separation. It follows from an intermediate result, also interesting by itself: for every integer t , the chromatic number χ can be bounded polynomially in terms of \mathbf{bp} if and only if it can be polynomially bounded in terms of \mathbf{bp}_t . We also introduce the notion of oriented bipartite packing, in which the Clique-Stable Set separation exactly translates. For instance, we show that the maximum fooling set of $CL - IS$ corresponds exactly to an oriented bipartite packing of the complete graph. Amano introduced an equivalent notion in [12].

In Section 5.4, we highlight links between the Clique-Stable Set separation problem and both the stubborn problem and 3-CCP. The quasi-dichotomy theorem for list- M partitions proceeds by covering all the solutions by $\mathcal{O}(n^{\log n})$ particular instances of 2-SAT, called 2-list assignments. A natural extension would be a covering of all the solutions with a polynomial number of 2-list assignments. We prove that the existence of a polynomial covering of all the maximal solutions (to be defined later) for the stubborn problem is equivalent to the existence of such a covering for all the solutions of 3-CCP, which in turn is equivalent to the $CL - IS$ problem.

5.2 Clique-Stable Set separation conjecture

The communication complexity problem $CL - IS$ can be formalized by a function $f : X \times Y \rightarrow \{0, 1\}$, where X is the set of cliques and Y the set of stable sets of a fixed graph G and $f(x, y) = 1$ if and only if x and y intersect. It can also be represented by a $|X| \times |Y|$ matrix M with $M_{x,y} = f(x, y)$. In the non-deterministic version, Alice is given a clique x , Bob is given a stable set y and a prover gives to both Alice and Bob a certificate of size $N^b(f)$, where $b \in \{0, 1\}$, in order to convince them that $f(x, y) = b$. Then, Alice and Bob exchange one final bit, saying whether they agree or disagree with the certificate.

The aim is to minimize $N^b(f)$ in the worst case. When x and y intersect on some vertex v , the prover can just provide v as a certificate, hence $N^1(f) = \mathcal{O}(\log n)$. The best upper bound so far on $N^0(f)$ is $\mathcal{O}(\log^2(n))$ [191], which actually is not better than the bound on the deterministic communication complexity.

A *combinatorial rectangle* $X' \times Y' \subseteq X \times Y$ is a subset of (possibly non-adjacent) rows X' and columns Y' of M . It is *b-monochromatic* if for all $(x, y) \in X' \times Y'$, $f(x, y) = b$. The minimum number

of b -monochromatic combinatorial rectangles needed to cover the b -inputs of M is denoted by $C^b(f)$ and verifies $N^b(f) = \lceil \log_2 C^b(f) \rceil$ [132]. A *fooling set* is a set \mathcal{F} of b -inputs of M such that for all $(x, y), (x', y') \in \mathcal{F}$, either $f(x', y) \neq b$ or $f(x, y') \neq b$. In other words, a fooling set is a set of b -inputs of M that cannot be pairwise contained into the same b -monochromatic rectangle. Hence, it provides a lower bound on $C^b(f)$. Given a 0-monochromatic rectangle $X' \times Y'$, one can construct a partition (A, B) by putting in A every vertex appearing in a clique of X' , and putting in B every vertex appearing in a stable set of Y' . There is no conflict doing this since no clique in X' intersects any stable set in Y' . We then extend (A, B) into a partition of the vertices by arbitrarily putting the other vertices into A . Observe that (A, B) separates every clique in X' from every stable set in Y' . Conversely, a partition that separates some cliques from some stable sets can be interpreted as a 0-monochromatic rectangle. Thus finding $C^0(f)$ (or, equivalently $N^0(f)$) is equivalent to finding the minimum number of cuts which separate all the cliques and the stable sets. In particular, there is a $\mathcal{O}(\log n)$ certificate for the $CL-IS$ problem if and only if there is a polynomial number of partitions separating all the cliques and the stable sets.

fooling set

A *cut* is a pair (A, B) such that $A \cup B = V$ and $A \cap B = \emptyset$. It *separates* a clique K and a stable set S if $K \subseteq A$ and $S \subseteq B$. Note that a clique and a stable set can be separated if and only if they do not intersect. Let \mathcal{K}_G be the set of cliques of G and \mathcal{S}_G be the set of stable sets of G . We say that a family \mathcal{F} of cuts is a *CS-separator* if for all $(K, S) \in \mathcal{K}_G \times \mathcal{S}_G$ which do not intersect, there exists a cut in \mathcal{F} that separates K and S . While it is generally believed that the following question is false, we state it in a positive way:

CS-separator

Conjecture 9 (Clique-Stable Set separation Conjecture). *There is a polynomial Q , such that for every graph G on n vertices, there is a CS-separator of size at most $Q(n)$.*

Note that in ??, Yannakakis conjectured that Conjecture 9 is not correct. On the opposite, Lovasz conjectured that it is correct for perfect graphs. Since we only give partial positive results and give equivalence between conjectures, we state it in the positive form. A first very easy result is that we can only focus on maximal cliques and stable sets.

Proposition 5.1. *Conjecture 9 holds if and only if a polynomial family \mathcal{F} of cuts separates all the maximal (in the sense of inclusion) cliques from the maximal stable sets that do not intersect.*

Proof. First note that one direction is direct. Let us prove the other one. Assume \mathcal{F} is a polynomial family that separates all the maximal cliques from the maximal stable sets that do not intersect. Let $Cut_{1,x}$ be the cut $(N[x], N^C[x])$ and $Cut_{2,x}$ be the cut $(N(x), N^C(x))$. Let us prove that $\mathcal{F}' = \mathcal{F} \cup \{Cut_{1,x} | x \in V\} \cup \{Cut_{2,x} | x \in V\}$ is a CS-separator.

Let (K, S) be a pair of clique and stable set. Extend K and S by adding vertices to get a maximal clique K' and a maximal stable set S' . Either K' and S' do not intersect, and there is a cut in \mathcal{F} that separates K' from S' (thus K from S). Or K' and S' intersect in x (recall that a clique and a stable set intersect on at most one vertex): if $x \in K$, then $Cut_{1,x}$ separates K from S , otherwise $Cut_{2,x}$ does. \square

Some classes of graphs have a polynomial CS-separator, this is for instance the case when \mathcal{C} is a class of graphs with a polynomial number of maximal cliques (we just cut every maximal clique from the rest of the graph). For example, chordal graphs have a linear number of maximal cliques

corresponding to nodes of the so-called clique-tree decomposition. A generalization due to Alekseev [3] asserts that the graphs without induced cycle of length four have a quadratic number of maximal cliques.

In this part, we first prove that random graphs have a polynomial CS-separator. Then we focus on classes on graph with a specific forbidden graph: more precisely, split-free graphs and graphs with no long paths nor antipaths. Conjecture 9 is unlikely to be true in the general case, however we believe it may be true on perfect graphs and more generally in the following setting:

Conjecture 10. *Let H be a fixed graph. Then the Clique-Stable Set separation conjecture is true on H -free graphs.*

5.2.1 Random graphs

Recall that the random graph $G(n, p)$ is a probability space over the set of graphs on the vertex set $\{1, \dots, n\}$ determined by $\Pr[ij \in E] = p$, with these events mutually independent. We say that $G(n, p)$ has clique number ω if ω satisfies $\mathbb{E}(\text{number of cliques of size } \omega) = 1$.

A family \mathcal{F} of cuts on a graph G with n vertices is a *complete (a, b) -separator* if for every pair (A, B) of disjoint subsets of vertices with $|A| \leq a$, $|B| \leq b$, there exists a cut $(U, V \setminus U) \in \mathcal{F}$ separating A and B , namely $A \subseteq U$ and $B \subseteq V \setminus U$. We say that $G(n, p)$ has a *polynomial complete (a, b) -separator* if there exists a polynomial P such that for all $p \in [0, 1]$, there exists a complete (a, b) -separator of size $P(n)$ in $G(n, p)$ with high probability.

Theorem 5.2. *$G(n, p)$ has an $\mathcal{O}(n^7)$ complete (ω, α) -separator where ω and α are respectively the clique number and the independence number of $G(n, p)$.*

Sketch of proof. Let $b = 1/p$ and $b' = 1/(1-p)$. The independence number and clique number of $G(n, p)$ are given by the following formulas, depending on p (see [25]):

$$\begin{aligned}\omega &= 2\log_b(n) - 2\log_b(\log_b n) + 2\log_b(e/2) + 1 + o(1) \\ \alpha &= 2\log_{b'}(n) - 2\log_{b'}(\log_{b'} n) + 2\log_{b'}(e/2) + 1 + o(1)\end{aligned}$$

Draw a random partition (V_1, V_2) where each vertex is put in V_1 independently from the others with probability p , and put in V_2 otherwise. Let (K, S) be a pair of a clique and a stable set of the graph which do not intersect. There are at most 4^n such pairs. The probability that $K \subseteq V_1$ and $S \subseteq V_2$ is at least $p^\omega(1-p)^\alpha$. Assume for a while that $p^\omega(1-p)^\alpha \geq 1/n^6$. Then (K, S) is separated by at least $1/n^6$ of all the partitions. By double counting, there exists a partition that separates at least $1/n^6$ of all the pairs. We delete these separated pairs, and there remain at most $(1 - 1/n^6) \cdot 4^n$ pairs. The same probability for a pair (K, S) to be cut by a random partition still holds, hence we can iterate the process k times until $(1 - 1/n^6)^k \cdot 4^n \leq 1$. This is satisfied for $k = 2n^7$ which is a polynomial in n . Thus there is a complete (ω, α) -separator of size polynomial in n .

The proof that $p^\omega(1-p)^\alpha \geq 1/n^6$ is just calculus starting from the formulas given at the beginning of the proof for α and ω , and can be found in details in [37]. For example, if $p = 1/2$ then $\omega = 2\log(n) + o(\log n)$ and $\alpha = 2\log(n) + o(\log n)$. Thus $p^\omega(1-p)^\alpha = 1/2^{4\log n + o(\log n)} = n^{4+o(1)}$. \square

Note here that no optimization was made on the constant of the polynomial. Some refinements in the proof can lead to a complete (ω, α) -separator of size $\mathcal{O}(n^{6+\epsilon})$. Moreover, an interesting ques-

tion would be a lower bound on the constant of the polynomial needed to separate the cliques and the stable sets in random graphs, in particular for the special case $p = 1/2$.

5.2.2 The case of split-free graphs.

Recall that graph Γ is split if its vertex set can be partitioned into a clique and a stable set and that a graph $G = (V, E)$ has an induced Γ if there exists $X \subseteq V$ such that the induced graph $G[X]$ is isomorphic to Γ . We denote by \mathcal{C}_Γ the class of graphs with no induced Γ . For instance, if Γ is a triangle with three pending edges, (see Figure 3.9), then \mathcal{C}_Γ contains the class of comparability graphs, for which Lovász showed [139] the existence of a CS-separator of size $\mathcal{O}(n^2)$. Our goal in this part is to prove that \mathcal{C}_Γ has a polynomial CS-separator when Γ is a split graph. The proof of the following result is based on a VC-dimensional argument.

Theorem 5.3. *Let Γ be a fixed split graph. Then the Clique-Stable Set conjecture is satisfied on \mathcal{C}_Γ .*

Proof. The vertices of Γ are partitioned into (V_1, V_2) where V_1 is a clique and V_2 is a stable set. Let $\varphi = \max(|V_1|, |V_2|)$ and $t = 64\varphi(\log(\varphi) + 2)$. Let $G = (V, E) \in \mathcal{C}_\Gamma$ and \mathcal{F} be the following family of cuts. For every clique $\{x_1, \dots, x_r\}$ with $r \leq t$, we note $U = \bigcap_{1 \leq i \leq r} N[x_i]$ and put $(U, V \setminus U)$ in \mathcal{F} . Similarly, for every stable set $\{x_1, \dots, x_r\}$ with $r \leq t$, we note $U = \bigcup_{1 \leq i \leq r} N(x_i)$ and put $(U, V \setminus U)$ in \mathcal{F} . Since each member of \mathcal{F} is defined with a set of at most t vertices, the size of \mathcal{F} is at most $\mathcal{O}(n^t)$. Let us now prove that \mathcal{F} is a CS-separator. Let (K, S) be a pair of maximal clique and stable set. We build H a hypergraph with vertex set K . For all $x \in S$, build the hyperedge $K \setminus N_G(x)$ (see Figure 5.4). Symmetrically, build H' a hypergraph with vertex set S . For all $x \in K$, build the hyperedge $S \cap N_G(x)$. The goal is to prove thanks to Theorem 3.23 that H or H' has bounded transversality τ . This will enable us to prove that (C, S) is separated by \mathcal{F} .

To begin with, let us introduce an auxiliary oriented graph B with vertex set $K \cup S$. For all $x \in K$ and $y \in S$, put the arc xy if $xy \in E$, and put the arc yx otherwise (see Figure 5.3). Given a set X of vertices and a weight function¹ f , we have $f(X) = \sum_{x \in X} f(x)$.

Lemma 5.4. *In B , there exists:*

- (i) *either a weight function $w : K \rightarrow \mathbb{R}^+$ such that $w(K) = 2$ and $\forall x \in S, w(N^+(x)) \geq 1$.*
- (ii) *or a weight function $w : S \rightarrow \mathbb{R}^+$ such that $w(S) = 2$ and $\forall x \in K, w(N^+(x)) \geq 1$.*

Proof. The proof is derived from an application of Lemma 2.22 presented in Chapter 2. Lemma 2.22 ensures that any directed graphs admits a weight function w of total weight one such that any vertex x satisfies $w(N^+(x)) \geq w(N^-(x))$. If $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, we note $x \neq 0$ if there exists i such that $x_i \neq 0$ and we note $x \geq 0$ if for every $i, x_i \geq 0$.

Apply Lemma 2.22 to B to obtain a weight function $w' : V \rightarrow [0, 1]$. Then $w'(V) = 1$, so either $w'(K) > 0$ or $w'(S) > 0$. Assume $w'(K) > 0$ (the other case is handled symmetrically). Consider the new weight function w defined by $w(x) = 2w'(x)/w'(K)$ if $x \in K$, and 0 otherwise. Then for all $x \in S$, on one hand $w(N^+(x)) \geq w(N^-(x))$ by extension of the property of w' , and on the other hand, $N^+(x) \cup N^-(x) = K$ by construction of B . Thus $w(N^+(x)) \geq w(K)/2 = 1$ since $w(K) = 2$. \square

1. Recall that a weight function is a non negative function.

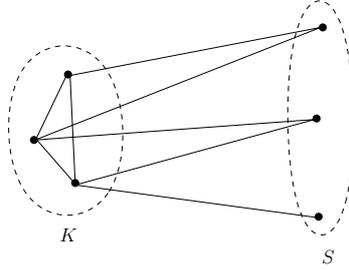
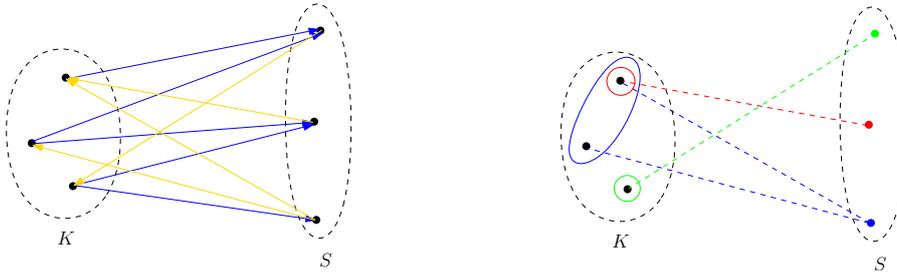
Figure 5.2: A clique K and a stable S in G .

Figure 5.3: Graph B built from K and S . Figure 5.4: Hypergraph H where hyperedges are built from the non-neighborhood of vertices from S . Edges of G are replaced by forward arcs, and non-edges are replaced by backward arcs.

In the following, let assume we are in case (i) and let us prove that H has bounded transversality. Case (ii) is handled symmetrically by switching H and H' .

Lemma 5.5. *The hypergraph H has fractional transversality $\tau^* \leq 2$.*

Proof. Let us prove that the weight function w given by Lemma 5.4 provides a solution to the fractional transversality linear program. Let e be a hyperedge built from the non-neighborhood of $x \in S$. Recall that this non-neighborhood is precisely $N^+(x)$ in B , then we have:

$$\sum_{y \in e} w(y) = w(N^+(x)) \geq 1.$$

Thus w satisfies the constraints of the fractional transversality, and $w(K) \leq 2$, *i.e.* we have $\tau^* \leq 2$. \square

Lemma 5.6. *H has VC-dimension bounded by $2\varphi - 1$.*

Proof. The proof is inspired from the proof of Lemma 3.9. Since it is slightly distinct, we nevertheless reprove it for the sake of completeness. Assume there is a set $A = \{u_1, \dots, u_\varphi, v_1, \dots, v_\varphi\}$ of 2φ vertices of H such that for every $B \subseteq A$ there is an edge $e \in E$ so that $e \cap A = B$. The aim is to exploit the shattering to find an induced Γ , which builds a contradiction. Recall that the forbidden split graph Γ is the union of a clique $V_1 = \{x_1, \dots, x_r\}$ and a stable set $V_2 = \{y_1, \dots, y_{r'}\}$ (with $r, r' \leq \varphi$). Let $x_i \in V_1$, let $\{y_{i_1}, \dots, y_{i_k}\} = N_\Gamma(x_i) \cap V_2$ be the set of its neighbors in V_2 .

Consider $\mathcal{U}_i = \{u_{i_1}, \dots, u_{i_k}\} \cup \{v_i\}$ (possible because $|V_1|, |V_2| \leq \varphi$). By assumption on A , there exists $e \in E$ such that $e \cap A = A \setminus \mathcal{U}_i$. Let $s_i \in S$ be the vertex whose non-neighborhood corresponds

to the edge e , then the neighborhood of s_i in A is exactly \mathcal{U}_i . Let $\mathcal{U} = \{u_1, \dots, u_\varphi\}$. Now, forget about the existence of v_1, \dots, v_φ , and observe that $N_G(s_i) \cap \mathcal{U} = \{u_{i_1}, \dots, u_{i_k}\}$. Then $G[\{s_1, \dots, s_r\} \cup \mathcal{U}]$ is an induced Γ , which is a contradiction. \square

As for Lemma 3.9, one can remark that a better analysis provides a $\varphi + \log \varphi$ upper bound.

Applying Theorem 3.23, and Lemma 5.5 and 5.6 to H , we obtain

$$\tau(H) \leq 16d\tau^*(H) \log(d\tau^*(H)) \leq 64\varphi(\log(\varphi) + 2) = t.$$

Hence τ is bounded by t which only depends on H . There must be $x_1, \dots, x_\tau \in K$ such that each hyperedge of H contains at least one x_i . Consequently, $S \subseteq \cup_{1 \leq i \leq \tau} N_G^C[x_i]$. Moreover, $K \subseteq (\cap_{1 \leq i \leq \tau} N_G[x_i]) = U$ since x_1, \dots, x_τ are in the same clique K . This means that the cut $(U, V \setminus U) \in \mathcal{F}$ built from the clique x_1, \dots, x_τ separates K and S .

When case (ii) of Claim 5.4 occurs, H' has bounded transversality, so there are τ vertices $x_1, \dots, x_\tau \in S$ such that for all $y \in K$, there exists $x_i \in N(y)$. Thus $K \subseteq (\cup_{1 \leq i \leq \tau} N_G(x_i)) = U$ and $S \subseteq \cap_{1 \leq i \leq \tau} N_G^C(x_i)$. The cut $(U, V \setminus U) \in \mathcal{F}$ built from the stable set x_1, \dots, x_τ separates K and S . \square

5.2.3 The case of $P_k, \overline{P_k}$ -free graphs

In this Section we prove that Conjecture 9 also holds for $(P_k, \overline{P_k})$ -free graphs. In order to prove it, we prove a more difficult result since we prove that the Erdős-Hajnal conjecture holds for $P_k, \overline{P_k}$ -free graphs. This proof will implies that Conjecture 9 holds for $(P_k, \overline{P_k})$ -free graphs. Before entering into the details, let us give some definitions on the Erdős-Hajnal conjecture and make a brief state of the art on this conjecture

A class \mathcal{C} of graphs (*i.e.* closed under induced subgraphs) is said to satisfy the *Erdős-Hajnal property* if there exists some $c > 0$ such that every graph on n vertices of \mathcal{C} contains a clique or a stable set of size n^c . The Erdős-Hajnal conjecture [80] asserts that every strict class of graphs satisfies the Erdős-Hajnal property, see [56] for a survey. This question is even open for graphs not inducing a C_5 . When excluding a single graph H , Alon, Pach and Solymosi showed in [10] that the conjecture holds if and only if it holds for every prime graph H (*i.e.* graph without nontrivial modules). A natural approach is then to study classes of graphs with intermediate difficulty, hoping to get a proof scheme which could be extended. A natural prime candidate to forbid is certainly the path. Chudnovsky and Zwols studied the class \mathcal{C}_k of graphs not inducing the path P_k on k vertices nor $\overline{P_k}$. They proved the Erdős-Hajnal property for P_5 and $\overline{P_6}$ -free graphs [59]. This was extended for P_5 and $\overline{P_7}$ -free graphs by Chudnovsky and Seymour [58]. We show that for every fixed k , the class \mathcal{C}_k satisfies the Erdős-Hajnal property and we extend it for obtaining Conjecture 9 for the class \mathcal{C}_k . A graph on n vertices is an ε -stable set if it has at most $\varepsilon \cdot \binom{n}{2}$ edges. The complement of an ε -stable set is an ε -clique. A stronger version of the following result was proved by Rödl [174]:

Erdős-
Hajnal

Theorem 5.7. *For every positive integer k and every $\varepsilon > 0$, there exists $\delta > 0$ such that every graph on n vertices G satisfies one of the following:*

- G induces all graphs on k vertices.
- G contains an ε -stable set of size at least δn .
- G contains an ε -clique of size at least δn .

Sketch of the proof. Note that by choosing δ small enough, we can assume that G is arbitrarily large, since a single vertex is certainly an ε -stable set. Let k' and k'' be two integers such that $\max(k, 2/\varepsilon) \ll k' \ll k''$. Select now some $\varepsilon' > 0$ such that $\varepsilon' \ll \min((1/k'')^2, (\varepsilon/2)^k)$. Applying Szemerédi's lemma, there exists M for which every large enough graph has an ε' -regular partition with at least k'' parts and at most M parts. We consider such an ε' -regular partition P of G . Since ε' is small enough, there are k'' classes of P forming pairwise ε' -regular pairs. Since $k'' \gg k'$ (precisely, providing that k'' is at least $\text{Ramsey}(k', k', k')$), we can find k' of them with all pairwise densities, either less than $\varepsilon/2$, or between $\varepsilon/2$ and $1 - \varepsilon/2$, or more than $1 - \varepsilon/2$. The first case gives an ε -stable set of size δn , where δ is k'/M . The last case gives an ε -clique of size δn . Since $k \leq k'$, the intermediate case provides k parts such that all pairs are ε' -regular with densities between $\varepsilon/2$ and $1 - \varepsilon/2$. Thus one can induce all possible graphs by choosing one vertex in each of the parts since $\varepsilon' \ll (\varepsilon/2)^k$. \square

Note that another proof of Theorem 5.7 was then provided by Fox and Sudakov [97], with no use of the regularity lemma, consequently giving a better constant $\delta = 2^{-15k(\log(1/\varepsilon))^2}$.

In a graph G , a *complete ℓ -bipartite graph* is a pair of disjoint subsets X, Y of vertices of G , both of size ℓ and inducing all edges between X and Y . We define similarly *empty ℓ -bipartite graph* when there is no edge between X and Y . Observe that we do not require any condition inside X or inside Y . Erdős, Hajnal and Pach proved in [81] that for every strict class \mathcal{C} , there exists some $c > 0$ such that every graph on n vertices in \mathcal{C} contains an empty or complete n^c -bipartite graph. This "half" version of the conjecture was improved to a "three quarter" version by Fox and Sudakov [98], where they show the existence of a polynomial clique or empty bipartite graph. It was proved that getting linear complete or empty bipartite graphs is enough to prove the full version:

Theorem 5.8 ([9, 96]). *If \mathcal{C} is a class of graphs for which there exists $c > 0$ such that every graph G of \mathcal{C} has an empty or complete $c \cdot n$ -bipartite graph, then \mathcal{C} satisfies the Erdős-Hajnal property.*

Sketch of the proof. Let $c' > 0$ such that $c'^c \geq 1/2$. We prove by induction that every graph G of \mathcal{C} induces a P_4 -free graph of size $n^{c'}$. By our hypothesis on \mathcal{C} , up to symmetry there exists a complete $c \cdot n$ -bipartite graph X, Y in G . Applying the induction hypothesis inside both X and Y , we form a P_4 -free graph on $2(c \cdot n)^{c'} \geq n^{c'}$ vertices. The Erdős-Hajnal property of \mathcal{C} follows from the fact that every P_4 -free graph on $n^{c'}$ vertices has a clique or a stable set of size at least $n^{c'/2}$. \square

Before going to the main result of this section, we state an intermediate result. If $d(x)$ is the degree of the vertex x , its *closed degree* $\overline{d}(x)$ is $d(x) + 1$, corresponding to the size of its closed neighborhood $N[x]$, that is to say $\{x\} \cup \{y \mid xy \in E\}$.

Theorem 5.9. *Let $n \geq 2$ be an integer, $c, \varepsilon \in [0, 1]$ and G be a connected graph on n vertices. Assume that G contains no empty $c \cdot n$ -bipartite subgraph and that every vertex x has closed degree $\overline{d}(x) \leq \varepsilon \cdot n$. Then for every vertex x , there exists a path P_k starting in x with $k \geq \frac{1}{2(\varepsilon+c)}$.*

Proof. We follow the lines of Gyárfás' proof of the χ -boundedness of P_k -free graphs, see [113]. First assume that $3c + \varepsilon \geq 1$. In particular, $\frac{1}{2(\varepsilon+c)} \leq \frac{3}{2}$. To conclude, observe that every vertex is incident to a P_2 (an edge) in a connected graph of size at least 2. From now on, assume that $3c + \varepsilon < 1$. Let x be a vertex, U be the set $V \setminus N[x]$, and C_1 be the largest connected component of $G[U]$. We distinguish three cases:

- if $|C_1| \geq (1 - \varepsilon - c) \cdot n$: let $y \in N[x]$ having a neighbor in C_1 , and $G' = G[\{y\} \cup C_1]$. Apply the induction hypothesis to G' containing no empty $c' \cdot n'$ -bipartite subgraph with $\overline{n'} = |G'| \geq (1 - \varepsilon - c) \cdot n$ and $c' = c/(1 - \varepsilon - c)$. Moreover, every vertex x of G' has closed degree $\overline{d(x)} \leq \varepsilon' \cdot n'$ with $\varepsilon' = \varepsilon/(1 - \varepsilon - c)$. Then, by induction, there exists a path $P_{k'}$ starting in y with $k' \geq \frac{1}{2(\varepsilon' + c')} = \frac{1}{2(\varepsilon + c)} - \frac{1}{2}$. Augmenting the path by x gives a path of length $k' + 1 \geq \frac{1}{2(\varepsilon + c)}$.
- if $c \cdot n \leq |C_1| \leq (1 - \varepsilon - c) \cdot n$: then $(C_1, U \setminus C_1)$ is an empty $c \cdot n$ -bipartite graph, a contradiction.
- otherwise $|C_1| \leq c \cdot n$: then U is divided into small connected components that we can split into two parts, each of size at least $c \cdot n$ (this is possible because the total size of U is at least $3c \cdot n$, due to $3c + \varepsilon < 1$). This forms an empty $c \cdot n$ -bipartite graph, a contradiction. □

We now have all the tools for proving the two main results of this Section:

Theorem 5.10. *For every k , there is a $c_k > 0$, such that every graph in \mathcal{C}_k contains an empty or complete bipartite graph on $c_k \cdot n$. Thus, by Theorem 5.8, the class \mathcal{C}_k satisfies the Erdős-Hajnal property.*

Proof. Let $\varepsilon > 0$ be some small value. By Theorem 5.7, there exists $\delta > 0$ such that every graph G not inducing P_k or \overline{P}_k does contain an ε -stable set or an ε -clique of size at least δn . Free to consider the complement of G , we can assume that G contains an ε -stable set S of size δn . We start by deleting in S all the vertices with degree in S more than $2\varepsilon s$ where s is the size of S . Since the average degree in S is at most εs , we do not delete more than half of the vertices. We still call S the remaining subgraph which is a 2ε -stable set of size $s \geq \delta n/2$ with maximum degree at most $2\varepsilon s$. Let $c \in [0, 1]$ be a small constant. If S contains no empty $c \cdot s$ -bipartite subgraph, we apply Theorem 5.9 and get a path $P_{k'}$ with $k' \geq \frac{1}{2(2\varepsilon + c)}$. Choosing ε and c small enough leads to a contradiction to G not inducing a P_k . Thus G contains an empty $c\delta/2 \cdot n$ -bipartite graph. □

Theorem 5.10 implies the clique-stable set separation for the class \mathcal{C}_k .

Theorem 5.11. *Let $k > 0$. The Clique-Stable set conjecture is satisfied on \mathcal{C}_k .*

Proof. The goal is to prove that every graph in \mathcal{C}_k admits a CS-separator of size n^c where $c = (-1/\log_2(1 - c_k))$. We proceed by contradiction and assume that G is a minimal counter-example. Free to exchange G and its complement, by Theorem 5.10, there exists two subsets V_1, V_2 completely non adjacent, and $|V_1|, |V_2| \geq c_k \cdot n$ for some constant $0 < c_k < 1$. Call $V_3 = V \setminus (V_1 \cup V_2)$. By minimality of G , $G[V_1 \cup V_3]$ admits a CS-separator F_1 of size $(|V_1| + |V_3|)^c$, and $G[V_2 \cup V_3]$ admits a CS-separator F_2 of size $(|V_2| + |V_3|)^c$. Let us build F aiming at being a CS-separator for G . For every cut (U, W) in F_1 , build the cut $(U, W \cup V_2)$, and similarly for every cut (U, W) in F_2 , build the cut $(U, W \cup V_1)$. We show that F is indeed a CS-separator: let (K, S) be a pair of clique and stable set of G that do not intersect, then either $K \subseteq V_1 \cup V_3$, or $K \subseteq V_2 \cup V_3$ since there is no edge between V_1 and V_2 . By symmetry, suppose $K \subseteq V_1 \cup V_3$, then there exists a cut (U, W) in F_1 that separates $(K, S \cap (V_1 \cup V_3))$ and the corresponding cut $(U, W \cup V_2)$ in F separates (K, S) . Finally, F has size at most $2 \cdot ((1 - c_k)n)^c \leq n^c$. □

5.2.4 Other classes of graphs

Lagoutte and Trunck recently proved in [135] that Conjecture 9 holds on perfect graphs with no balanced skew partition. It gives some evidence on the Lovasz's conjecture which states that Conjecture 9 holds for perfect graphs. Though, one can note that the main complexity of lots of proofs on perfect graphs is due to skew partitions. So there might be a complexity gap between their proof and the proof of the whole conjecture.

5.3 Bipartite packing and graph coloring

The goal of this section is to prove that the polynomial Alon-Saks-Seymour conjecture is equivalent to the Clique-Stable Set separation conjecture. We need for this an intermediate step using a new version of the Alon-Saks-Seymour conjecture, called the Oriented Alon-Saks-Seymour conjecture.

5.3.1 Oriented Alon-Saks-Seymour conjecture

$\mathbf{bp}(G)$

The *bipartite packing* $\mathbf{bp}(G)$ of a graph G is the minimum number of edge-disjoint complete bipartite graphs needed to partition the edges of G . Alon, Saks and Seymour conjectured that if $\mathbf{bp}(G) \leq k$, then $\chi(G) \leq k + 1$. The conjecture holds for complete graphs. Indeed, Graham and Pollak [107] proved that $n - 1$ edge-disjoint complete bipartite graphs are needed to partition the edges of K_n . A beautiful algebraic proof of this theorem is due to Tverberg [188]. The conjecture was disproved by Huang and Sudakov in [120] who proved that $\chi \geq k^{6/5}$ for some graphs using a construction based on Razborov's graphs [170]. It was improved into $\chi \geq k^{3/2}$ by Amano in [12]. Nevertheless the existence of a polynomial bound is still open.

Conjecture 11 (Polynomial Alon-Saks-Seymour Conjecture). *There exists a polynomial P such that for every G , $\chi(G) \leq P(\mathbf{bp}(G))$.*

Note that Huang and Sudakov conjectured the converse of Conjecture 11. We nevertheless decided to state all the conjectures on their positive version. We introduce a variant of the bipartite packing which may lead to a new superlinear lower bound on the Clique-Stable separation. The *oriented bipartite packing* $\mathbf{bp}_{\text{or}}(G)$ of a non-oriented graph G is the minimum number of oriented complete bipartite graphs such that each edge is covered by an arc in at least one direction (it can be in both directions), but it cannot be covered twice in the same direction (see Figure 5.5 for an example). Note that Amano recently introduced an equivalent version of this object in [12]. He called these partitions ordered biclique partitions. A *packing certificate of size k* is a set $\{(A_1, B_1), \dots, (A_k, B_k)\}$ of k oriented bipartite subgraphs of G that fulfill the above conditions restated as follows: for each edge xy of G , free to exchange x and y , there exists i such that $x \in A_i, y \in B_i$, but there do not exist distinct i and j such that $x \in A_i \cap A_j$ and $y \in B_i \cap B_j$.

Conjecture 12 (Oriented Alon-Saks-Seymour Conjecture). *There exists a polynomial P such that for every G , $\chi(G) \leq P(\mathbf{bp}_{\text{or}}(G))$.*

First of all, we prove that studying $\mathbf{bp}_{\text{or}}(K_m)$ is deeply linked with the existence of a fooling set for $CL - IS$. Recall the definitions of Section 5.2: in the communication matrix M for $CL - IS$,

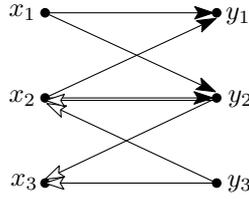


Figure 5.5: A graph G such that $\mathbf{bp}_{\text{or}}(G) = 2$ (and $\mathbf{bp}(G) = 3$). Two different kinds of arrows show a packing certificate of size 2: $(\{x_1, x_2\}, \{y_1, y_2\})$ and $(\{y_2, y_3\}, \{x_2, x_3\})$. The edge x_2y_2 is covered once in each direction, while the other edges are covered in exactly one direction.

each row corresponds to a clique K , each column corresponds to a stable set S , and $M_{K,S} = 1$ if K and S intersect, 0 otherwise. A *fooling set* \mathcal{C} is a set of pairs (K, S) such that K and S do not intersect, and for all $(K, S), (K', S') \in \mathcal{C}$, K intersects S' or K' intersects S (consequently $M_{K,S'} = 1$ or $M_{K',S} = 1$). Thus \mathcal{C} is a set of 0-entries of the matrix that pairwise can not be put together into the same combinatorial 0-rectangle. The maximum size of a fooling set consequently is a lower bound on the non-deterministic communication complexity for $CL - IS$, and consequently on the size of a CS-separator.

Theorem 5.12. *Let $n, m \in \mathbb{N}^*$. There exists a fooling set \mathcal{C} of size m on some graph on n vertices if and only if $\mathbf{bp}_{\text{or}}(K_m) \leq n$.*

Lemma 5.13, 5.14, 5.17 and 5.18 follow the scheme of proofs of Alon and Haviv which can be found in [120].

Lemma 5.13. *Let $n, m \in \mathbb{N}^*$. If there exists a fooling set \mathcal{C} of size m on some graph G on n vertices then $\mathbf{bp}_{\text{or}}(K_m) \leq n$.*

Proof. Consider all pairs (K, S) of cliques and stable set in the fooling set \mathcal{C} , and construct an auxiliary graph H in the same way as in the proof of Lemma 5.17: the vertices of H are the m pairs (K, S) of the fooling set and there is an edge between (K, S) and (K', S') if and only if there is a vertex in $S \cap K'$ or in $S' \cap K$. By definition of a fooling set, H is a complete graph. For $x \in V(G)$, let (A_x, B_x) be the oriented bipartite subgraph of H where A_x is the set of pairs (K, S) for which $x \in K$, and B_x is the set of pairs (K, S) for which $x \in S$. This defines a packing certificate of size n on H : first of all, by definition of the edges, (A_x, B_x) is complete. Moreover, every edge is covered by such a bipartite: if $(K, S)(K', S') \in E(H)$ then there exists $x \in S \cap K'$ or $x \in S' \cap K$ thus the corresponding arc is in (A_x, B_x) . Finally, an arc $(K, S)(K', S')$ can not appear in both (A_x, B_x) and (A_y, B_y) otherwise the stable set S and the clique K' intersect on two vertices x and y , which is impossible. Hence $\mathbf{bp}_{\text{or}}(H) \leq n$. H being a complete graph on m elements proves the lemma. \square

Lemma 5.14. *Let $n, m \in \mathbb{N}^*$. If $\mathbf{bp}_{\text{or}}(K_m) \leq n$ then there exists a fooling set of size m on some graph G on n vertices.*

Proof. Construct an auxiliary graph H : the vertices are the elements of a packing certificate of size n , and there is an edge between (A_1, B_1) and (A_2, B_2) if and only if there is a vertex $x \in A_1 \cap A_2$. Then for all $x \in V(K_m)$, the set of all bipartite graphs (A, B) with $x \in A$ form a clique called K_x , and

the set of all bipartite graphs (A, B) with $x \in B$ form a stable set called S_x . S_x is indeed a stable set, otherwise there are (A_1, B_1) and (A_2, B_2) in S_x (implying $x \in B_1 \cap B_2$) linked by an edge resulting from a vertex $y \in A_1 \cap A_2$, then the arc yx is covered twice. Consider all pairs (K_x, S_x) for $x \in V(K_m)$: this is a fooling set of size m . Indeed, on one hand $K_x \cap S_x = \emptyset$. On the other hand, for all $x, y \in V(K_m)$, the edge xy is covered by a complete bipartite graph (A, B) with $x \in A$ and $y \in B$ (or conversely). Then K_x and S_y (or K_y and S_x) intersects in (A, B) . \square

Proof of Theorem 5.12. Lemmas 5.13 and 5.14 conclude the proof. \square

One can search for an algebraic lower bound for $\mathbf{bp}_{\text{or}}(K_m)$. Let $(A_1, B_1), \dots, (A_k, B_k)$ be a packing certificate of K_m . For every i construct the $m \times m$ matrix M^i such that $M^i_{u,v} = 1$ if $u \in A_i, v \in B_i$ and 0 otherwise, then M^i has rank 1. Let $M = \sum_{i=1}^k M^i$, then by construction M has rank at most k , and has the three following particularities: it contains only 0 and 1, its diagonal entries are all 0, and for every distinct i, j , $M_{i,j} = 1$ or $M_{j,i} = 1$ (or both). This is due to the definition of a packing certificate. A natural question arising is to find a lower bound on the minimum rank of a $m \times m$ matrix respecting these three particularities. This will imply a lower bound on $\mathbf{bp}_{\text{or}}(K_m)$, and thus an upper bound on the size of a fooling set.

Theorem 5.12 implies that if $\mathbf{bp}_{\text{or}}(K_n) = \mathcal{O}(n^{1/k})$, then there exists a fooling set of size $\Omega(n^k)$ on some graphs G on n vertices, thus $\Omega(n^k)$ is a lower bound on the Clique-Stable Set separation. Yeo [192] proved that $\mathbf{bp}_{\text{or}}(K_n) \leq \mathcal{O}(n/2\sqrt{\log n})$, but this bound was improved by Amano in [12] who proved $\mathbf{bp}_{\text{or}}(K_n) \leq \mathcal{O}(n^{2/3})$. The best lower bound is the following:

Observation 5.15. *Let G be a graph. Then there exists a fooling set \mathcal{F} on G of size $|V(G)| + 1$.*

Proof. Let us do the proof by induction on $|V(G)|$. If $V = \{v\}$, consider the clique $\{v\}$ together with the empty stable set, and the stable set $\{v\}$ together with the empty clique. This is a fooling set of size 2. If $|V| = n + 1$, let $v \in V$, $n_1 = |N(v)|$, $n_2 = |N^C[v]|$, with $n = n_1 + n_2 + 1$. Then the induction hypothesis gives a fooling set \mathcal{F}_1 of size $n_1 + 1$ on $N(v)$, and a fooling set \mathcal{F}_2 of size $n_2 + 1$ on $N^C[v]$. Extend each clique of \mathcal{F}_1 with v , which still forms a clique; and extend each stable set of \mathcal{F}_2 with v , which still forms a stable set. This gives a fooling set \mathcal{F} of size $n_1 + 1 + n_2 + 1 = n + 1$. It is indeed a fooling set: if $(K, S), (K', S') \in \mathcal{F}$, either they come both from \mathcal{F}_1 or both from \mathcal{F}_2 , so the property is satisfied by \mathcal{F}_1 and \mathcal{F}_2 being fooling sets; either (K, S) initially comes from \mathcal{F}_1 and (K', S') from \mathcal{F}_2 , and then $K \cap S' = \{v\}$. \square

In fact the oriented Alon-Saks-Seymour conjecture is equivalent to the Clique-Stable Set separation conjecture.

Theorem 5.16. *The oriented Alon-Saks-Seymour conjecture is satisfied if and only if the Clique-Stable Set separation conjecture is satisfied.*

The proof is very similar to the one of Theorem 5.12.

Lemma 5.17. *If the oriented Alon-Saks-Seymour conjecture is satisfied, then the Clique-Stable Set separation conjecture is satisfied.*

Proof. Let G be a graph on n vertices. We want to separate all the pairs of cliques and stable sets which do not intersect. Consider all the pairs (K, S) such that the clique K does not intersect the stable set S . Construct an auxiliary graph H as follows. The vertices of H are the pairs (K, S) and there is an edge between a pair (K, S) and a pair (K', S') if and only if there is a vertex $x \in S \cap K'$ or $x \in S' \cap K$. For every vertex x of G , let (A_x, B_x) be the oriented bipartite subgraph of H where A_x is the set of pairs (K, S) for which $x \in K$, and B_x is the set of pairs (K, S) for which $x \in S$. By definition of the edges, (A_x, B_x) is complete. Moreover, every edge is covered by such a bipartite: if $(K, S)(K', S') \in E(H)$ then there exists $x \in S \cap K'$ or $x \in S' \cap K$ thus the corresponding arc is in (A_x, B_x) . Finally, an arc $(K, S)(K', S')$ can not appear in both (A_x, B_x) and (A_y, B_y) otherwise the stable set S and the clique K' intersect on two vertices x and y , which is impossible. Hence the oriented bipartite packing of this graph is at most n .

If the oriented Alon-Saks-Seymour conjecture is satisfied, then $\chi(H) \leq P(n)$. Consider a color of this polynomial coloring. Let A be the set of vertices of this color, so A is a stable set. Then the union of all the second components (corresponding to stable sets of G) of the vertices of A do not intersect the union of all the first components (corresponding to cliques of G) of A . Otherwise, there are two vertices (K, S) and (K', S') of A such that K intersects S' , thus $(K, S)(K', S')$ is an edge. This is impossible since A is a stable set.

The union of the cliques of A and the union of the stable sets of A do not intersect, hence it defines a cut which separates all the pairs of A . The same can be done for every color. Then we can separate all the pairs (K, S) by $\chi(H) \leq P(n)$ cuts, which achieves the proof. \square

Lemma 5.18. *If the Clique-Stable Set separation conjecture is satisfied, then the oriented Alon-Saks-Seymour conjecture is satisfied.*

Proof. Let $G = (V, E)$ be a graph with $\mathbf{bp}_{\text{or}}(G) = k$. Construct an auxiliary graph H as follows. The vertices are the elements of a packing certificate of size k . There is an edge between two elements (A_1, B_1) and (A_2, B_2) if and only if there is a vertex $x \in A_1 \cap A_2$. Hence the set of all (A_i, B_i) such that $x \in A_i$ is a clique of H (say the clique K_x associated to x). The set of all (A_i, B_i) such that $y \in B_i$ is a stable set in H (say the stable set S_y associated to y). Indeed, if $y \in B_1 \cap B_2$ and there is an edge resulting from $x \in A_1 \cap A_2$, then the arc xy is covered twice which is impossible. Note that a clique or a stable set associated to a vertex can be empty, but this does not trigger any problem. Since the Clique-Stable set separation conjecture is satisfied, there are $P(k)$ (with P a polynomial) cuts which separate all the pairs (K, S) , in particular which separate all the pairs (K_x, S_x) for $x \in V$.

Associate to each cut a color, and let us now color the vertices of G with them. We color each vertex x by the color of the cut separating (K_x, S_x) . Let us finally prove that this coloring is proper. Assume there is an edge xy such that x and y are given the same color. Then there exists a bipartite graph (A, B) that covers the edge xy , hence (A, B) is in both K_x and S_y . Since x and y are given the same color, then the corresponding cut separates both K_x from S_x and K_y from S_y . This is impossible because K_x and S_y intersects in (A, B) . Then we have a coloring with at most $P(k)$ colors. \square

Proof of Theorem 5.16. This is straightforward using Lemmas 5.17 and 5.18. \square

5.3.2 Generalization: t -biclique covering numbers

We introduce here a natural generalization of the Alon-Saks-Seymour conjecture, studied by Huang and Sudakov in [120]. While the Alon-Saks-Seymour conjecture deals with partitioning the edges, we relax here to a covering of the edges by complete bipartite graphs, meaning that an edge can be covered several times. Formally, a t -biclique covering of an undirected graph G is a collection of complete bipartite graphs that covers every edge of G at least once and at most t times. The minimum size of such a covering is called the t -biclique covering number, and is denoted by $\mathbf{bp}_t(G)$. In particular, $\mathbf{bp}_1(G)$ is the usual bipartite packing $\mathbf{bp}(G)$.

In addition to being an interesting parameter to study in its own right, the t -biclique covering number of complete graphs is also closely related to a question in combinatorial geometry about neighborly families of boxes. It was studied by Zaks [193] and then by Alon [5], who proved that \mathbb{R}^d has a t -neighborly family of k standard boxes if and only if the complete graph K_k has a t -biclique covering of size d (see [120] for definitions and further details). Better bounds were given by Huang and Sudakov in [120] Alon also gives asymptotic bounds for $\mathbf{bp}_t(K_k)$:

$$(1 + o(1))(t!/2^t)^{1/t} k^{1/t} \leq \mathbf{bp}_t(K_k) \leq (1 + o(1))tk^{1/t} .$$

Our results are concerned not only with K_k but for every graph G . It is natural to ask the same question for $\mathbf{bp}_t(G)$ as for $\mathbf{bp}(G)$, namely:

Conjecture 13 (Generalized Alon-Saks-Seymour conjecture of order t). *There exists a polynomial P_t such that for all graphs G , $\chi(G) \leq P_t(\mathbf{bp}_t(G))$.*

A t -biclique covering is a fortiori a t' -biclique covering for all $t' \geq t$. Moreover, a packing certificate of size $\mathbf{bp}_{\text{or}}(G)$, which covers each edge at most once in each direction can be seen as a non-oriented biclique covering which covers each edge at most twice. Hence, we have the following inequalities:

Observation 5.19. *For every graph G :*

$$\dots \leq \mathbf{bp}_{t+1}(G) \leq \mathbf{bp}_t(G) \leq \mathbf{bp}_{t-1}(G) \leq \dots \leq \mathbf{bp}_2(G) \leq \mathbf{bp}_{\text{or}}(G) \leq \mathbf{bp}_1(G) .$$

Observation 5.19 and bounds on $\mathbf{bp}_2(K_n)$ [5] give $\mathbf{bp}_{\text{or}}(K_n) \geq \mathbf{bp}_2(K_n) \geq \Omega(\sqrt{n})$. Then Theorem 5.12 ensures that the maximal size of a fooling set on a graph on n vertices is $\mathcal{O}(n^2)$.

Theorem 5.20. *Let $t \in \mathbb{N}^*$. The generalized Alon-Saks-Seymour conjecture of order t holds if and only if it holds for order 1.*

Proof. Assume the generalized Alon-Saks-Seymour conjecture of order t holds. Then $\chi(G)$ is bounded by a polynomial in $\mathbf{bp}_t(G)$ and thus, according to Observation 5.19, by a polynomial in $\mathbf{bp}_1(G)$. Hence the generalized Alon-Saks-Seymour of order 1 holds.

Now we focus on the other direction, and assume that the generalized Alon-Saks-Seymour conjecture of order 1 holds. Let us prove the result by induction on t , initialization for $t = 1$ being obvious. Let $G = (V, E)$ be a graph and let $\mathcal{B} = (B_1, \dots, B_k)$ be a t -biclique covering. Then E can be partitioned into E_t the set of edges that are covered exactly t times in \mathcal{B} , and $E_{<t}$ the set of edges that are covered at most $t - 1$ times in \mathcal{B} . Construct an auxiliary graph H with the same vertex set V as G and with edge set E_t .

Claim 5.21. $\mathbf{bp}_1(H) \leq (2k)^t$.

Since the Alon-Saks-Seymour of order 1 holds, then there exists a polynomial P such that $\chi(H) \leq P((2k)^t)$. Consequently V can be partitioned into $(S_1, \dots, S_{P((2k)^t)})$ where S_i is a stable set in H . In particular, the induced graph $G[S_i]$ contains no edge of E_t . Consequently $(B_1 \cap S_i, \dots, B_k \cap S_i)$ is a $(t-1)$ biclique covering of $G[S_i]$, where $B_j \cap S_i$ is the bipartite graph B_j restricted to the vertices of S_i . Thus $\mathbf{bp}_{t-1}(G[S_i]) \leq k$. By induction hypothesis, the generalized Alon-Saks-Seymour of order $(t-1)$ holds, so there exists a polynomial P_{t-1} such that $\chi(G[S_i]) \leq P_{t-1}(k)$. Let us now color the vertices of G with at most $P((2k)^t) \cdot P_{t-1}(k)$ colors, which is a polynomial in k . Each vertex $v \in S_i$ is given color (α, β) , where α is the color of S_i in H and β is the color of x in $G[S_i]$. This is a proper coloring of G , thus the generalized Alon-Saks-Seymour conjecture of order t holds. \square

Proof of Claim 5.21. For each B_i , let (B_i^-, B_i^+) be its partition into a complete bipartite graph. We number x_1, \dots, x_n the vertices of H . Let $x_i x_j$ be an edge, with $i < j$, then $x_i x_j$ is covered by exactly t bipartite graphs B_{i_1}, \dots, B_{i_t} . We give to this edge the label $((B_{i_1}, \dots, B_{i_t}), (\varepsilon_1, \dots, \varepsilon_t))$, where $\varepsilon_l = -1$ if $x_i \in B_{i_l}^-$ (then $x_j \in B_{i_l}^+$) and $\varepsilon_l = +1$ otherwise (then $x_i \in B_{i_l}^+$ and $x_j \in B_{i_l}^-$). For each such label \mathcal{L} appearing in H , call $E_{\mathcal{L}}$ the set of edges labeled by \mathcal{L} and define a set of edges $B_{\mathcal{L}} = E(B_{i_1}) \cap E_{\mathcal{L}}$. Observe that $B_{\mathcal{L}}$ forms a bipartite graph. The goal is to prove that the set of every $B_{\mathcal{L}}$ is a 1-biclique covering of H . Since there can be at most $(2k)^t$ different labels, this will conclude the proof.

Let us first observe that each edge appears in exactly one $B_{\mathcal{L}}$ because each edge has exactly one label. Let \mathcal{L} be a label, and let us prove that $B_{\mathcal{L}}$ is a complete bipartite graph. If $x_i x_{i'} \in B_{\mathcal{L}}$ and $x_j x_{j'} \in B_{\mathcal{L}}$, with $i < i'$ and $j < j'$ then these two edges have the same label $\mathcal{L} = ((B_{i_1}, \dots, B_{i_t}), (\varepsilon_1, \dots, \varepsilon_t))$. If $\varepsilon_l = -1$ (the other case in handle symmetrically), then x_i and x_j are in $B_{i_l}^-$ and $x_{i'}$ and $x_{j'}$ are in $B_{i_l}^+$. As B_{i_l} is a complete bipartite graph, then the edges $x_i x_{i'}$ and $x_j x_{j'}$ appear in $E(B_{i_l})$. Thus these two edges have also the label \mathcal{L} , so they are in $B_{\mathcal{L}}$: as conclusion, $B_{\mathcal{L}}$ is a complete bipartite graph. \square

5.4 3-CCP and the stubborn problem

The following definitions are illustrated on Figure 5.9 and deal with list coloring. Let G be a graph and COL a set of k colors. A set of possible colors, called *constraint*, is associated to each vertex. If the set of possible colors is COL then the constraint on this vertex is *trivial*. A vertex has an *l-constraint* if its set of possible colors has size at most l . An *l-list assignment* is a function $\mathcal{L} : V \rightarrow \mathcal{P}(\text{COL})$ that gives each vertex an *l-constraint*. A solution \mathcal{S} is a coloring of the vertices $\mathcal{S} : V \rightarrow \text{COL}$ that respects some requirements depending on the problem. We can equivalently consider \mathcal{S} as a partition (A_1, \dots, A_k) of the vertices of the graph with $x \in A_i$ if and only if $\mathcal{S}(x) = A_i$ (by abuse of notation A_i denotes both the color and the set of vertices having this color). An *l-list assignment* \mathcal{L} is *compatible* with a solution \mathcal{S} if for each vertex x , $\mathcal{S}(x) \in \mathcal{L}(x)$. A set of *l-list assignment* *covers* a solution \mathcal{S} if at least one of the *l-list assignment* is compatible with \mathcal{S} .

We recall the definitions of 3-CCP and the stubborn problem:

3-COMPATIBLE COLORING PROBLEM (3-CCP)

Input: An edge coloring f_E of K_n with 3 colors $\{A, B, C\}$.

Question: Is there a coloring of the vertices with $\{A, B, C\}$, such that no edge has the same color as both its endpoints?

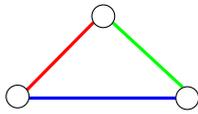


Figure 5.6: An instance of 3-CCP

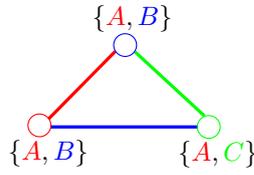


Figure 5.7: A solution to the instance (vertex coloring) together with a compatible 2-list assignment: each vertex has a 2-constraint

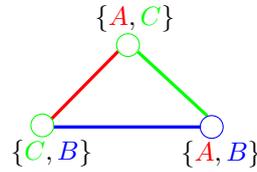


Figure 5.8: Another solution to the instance with a compatible 2-list assignment.

Figure 5.9: Illustration of definitions. Color correspondence: A=red ; B=blue ; C=green. Both 2-list assignments together form a 2-list covering because any solution is compatible with at least one of them.

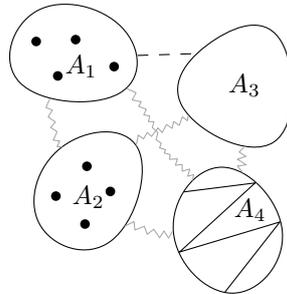


Figure 5.10: Diagram representing the stubborn problem. Cliques are represented by hatched sets, stable sets by dotted sets. Completely non-adjacent sets are linked by a dashed edge. Grey lines represent edges that may or may not appear in the graph.

STUBBORN PROBLEM

Input: A graph $G = (V, E)$ together with a list assignments $\mathcal{L} : V \rightarrow \mathcal{P}(\{A_1, A_2, A_3, A_4\})$.

Question: Can V be partitioned into four sets A_1, \dots, A_4 such that A_4 is a clique, both A_1 and A_2 are stable sets, A_1 and A_3 are completely non-adjacent, and the partition is compatible with \mathcal{L} ?

Given an edge-coloring f_E on K_n , a set of 2-list assignment is a *2-list covering for 3-CCP on (K_n, f_E)* if it covers all the solutions of 3-CCP on this instance. Moreover, 3-CCP is said to have a *polynomial 2-list covering* if there exists a polynomial P such that for every n and for every edge-coloring f_E , there is a 2-list covering on (K_n, f_E) whose cardinality is at most $P(n)$.

Symmetrically, we want to define a *2-list covering for the stubborn problem*. However, there is no hope to cover all the solutions of the stubborn problem on each instance with a polynomial number of 2-list assignments. Indeed if G is a stable set of size n and if every vertex has the trivial 4-constraint, then for any partition of the vertices into 3 sets (A_1, A_2, A_3) , there is a solution

$(A_1, A_2, A_3, \emptyset)$. Since there are 3^n partitions into 3 sets, and since every 2-list assignment covers at most 2^n solutions, all solutions cannot be covered with a polynomial number of 2-list assignments.

Thus we need a notion of maximal solutions. This notion is extracted from the notion of domination (here A_3 dominates A_1) in the language of general list- M partition problem (see [89]). Intuitively, if $\mathcal{L}(v)$ contains both A_1 and A_3 and v belongs to A_1 in some solution \mathcal{S} , we can build a simpler solution by putting v in A_3 and leaving everything else unchanged. A solution (A_1, A_2, A_3, A_4) of the stubborn problem on (G, \mathcal{L}) is a *maximal solution* if no member of A_1 satisfies $A_3 \in \mathcal{L}(v)$. We may note that if A_3 is contained in every $\mathcal{L}(v)$ for $v \in V$, then every maximal solution of the stubborn problem on (G, \mathcal{L}) let A_1 empty. Now, a set of 2-list assignments is a *2-list covering for the stubborn problem on (G, \mathcal{L})* if it covers all the maximal solutions on this instance. Moreover, it is called a *polynomial 2-list covering* if its size is bounded by a polynomial in the number of vertices in G .

For edge-colored graphs, an $(\alpha_1, \dots, \alpha_k)$ -*clique* is a clique for which every edge has a color in $\{\alpha_1, \dots, \alpha_k\}$. A *split* graph is the union of an α -clique and a β -clique. The α -*edge-neighborhood* of x is the set of vertices y such that xy is an α -edge, i.e an edge colored with α . The *majority color of $x \in V$* is the color α for which the α -edge-neighborhood of x is maximal in terms of cardinality (in case of ties, we arbitrarily cut them).

In this section, we prove that the existence of a polynomial 2-list covering for the stubborn problem is equivalent to the existence of a polynomial one for 3-CCP, which in turn is equivalent to the existence of a polynomial CS-separator. We first justify the interest of 2-list coverings.

Observation 5.22. *Given a 2-list assignment for 3-CCP, it is possible to decide in polynomial time if there exists a solution covered by it.*

Proof. Any 2-list assignment can be translated into an instance of 2-SAT. Each vertex has a 2-constraint $\{\alpha, \beta\}$ from which we construct two variables x_α and x_β and a clause $x_\alpha \vee x_\beta$. Turn x_α to true will mean that x is given the color α . Then we need also the clause $\neg x_\alpha \vee \neg x_\beta$ saying that only one color can be given to x . Finally for all edge xy colored with α , we add the clause $\neg x_\alpha \vee \neg y_\alpha$ if both variables exists, and no clause otherwise. \square

Therefore, given a polynomial 2-list covering, it is possible to decide in polynomial time if the instance of 3-CCP has a solution. Observe nevertheless that the existence of a polynomial 2-list covering does not imply the existence of a polynomial algorithm. Indeed, such a 2-list covering may not be computable in polynomial time.

Theorem 5.23 (Feder, Hell [87]). *There exists an algorithm giving a 2-list covering of size $\mathcal{O}(n^{\log n})$ for 3-CCP. By Observation 5.22, this gives an algorithm in time $\mathcal{O}(n^{\log n})$ which solves 3-CCP.*

Proof. Let us build a tree of maximum degree $n + 1$ and height $\mathcal{O}(\log n)$ whose leaves will exactly be the 2-list assignments needed to cover all the solutions. By a counting argument, such a tree will have at most $\mathcal{O}(n^{\log n})$ leaves, on which we can apply Observation 5.22 to have an algorithm in time $\mathcal{O}(n^{\log n})$ which solves 3-CCP.

Let x be a vertex, up to symmetry we can assume that x has majority color A . The solutions are partitioned between those where x is given its majority color A , and those where x is given color B or C . From this simple remark, we can build a tree with an unlabeled root, n children each labeled by a different vertex, and an extra leaf corresponding to the solutions where no vertex

is colored by its majority color. The latter forms a 2-list assignment since we forbid one color for each vertex. Each labeled child of the root, say its label is x , will consider only solutions where x is given its majority color A , thus x has constraint $\{A\}$. Then in every such solution, each vertex linked to x by an A -edge will be given the color B or C . Thus we associate the 2-constraint $\{B, C\}$ to the whole A -edge-neighborhood of x . Since the graph is complete and A is the majority color, this A -edge-neighborhood represents at least $1/3$ of all the vertices. We iterate the process on the graph restricted to unconstrained vertices, and build a subtree rooted at node x . We do so for the other labeled children of the root. The tree is ensured to have height $\mathcal{O}(\log n)$ because we erase at least $1/3$ of the vertices at each level. \square

Theorem 5.24. *The following are equivalent:*

1. *For every graph G and every list assignment $\mathcal{L} : V \rightarrow \mathcal{P}(\{A_1, A_2, A_3, A_4\})$, there is a polynomial 2-list covering for the stubborn problem on (G, \mathcal{L}) .*
2. *For every n and every edge-coloring $f : E(K_n) \rightarrow \{A, B, C\}$, there is a polynomial 2-list covering for 3-CCP on (K_n, f) .*
3. *For every graph G , there is a polynomial CS-separator.*

We decompose the proof into three lemmas, each of which describing one implication.

Lemma 5.25 ($1 \Rightarrow 2$). *Suppose for every graph G and every list assignment $\mathcal{L} : V \rightarrow \mathcal{P}(\{A_1, \dots, A_4\})$, there is a polynomial 2-list covering for the stubborn problem on (G, \mathcal{L}) . Then for every graph n and every edge-coloring $f : E(K_n) \rightarrow \{A, B, C\}$, there is a polynomial 2-list covering for 3-CCP on (K_n, f) .*

Proof. Let $n \in \mathbb{N}$, (K_n, f) be an instance of 3-CCP, and x a vertex of K_n . Let us build a polynomial number of 2-list assignments that cover all the solutions where x is given color A . Since the colors are symmetric, we just have to multiply the number of 2-list assignments by 3 to cover all the solutions. Let (A, B, C) be a solution of 3-CCP where $x \in A$.

Claim 5.26. *Let x be a vertex and α, β, γ be the three different colors. Let U be the α -edge-neighborhood of x . If there is a $\beta\gamma$ -clique Z of U which is not split, then there is no solution where x is colored with α .*

Proof. Consider a solution in which x is colored with α . All the vertices of Z are of color β or γ because they are in the α -edge-neighborhood of x . The vertices of Z colored with β form a γ -clique, those colored by γ form a β -clique. Hence Z is split. \square

A vertex x is *really 3-colorable* if for each color α , every $\beta\gamma$ -clique of the α -edge-neighborhood of x is a split graph. If a vertex is not really 3-colorable then, in a solution, it can be colored by at most 2 different colors. Hence if $K_n[V \setminus x]$ has a polynomial 2-list covering, the same holds for K_n by assigning the only two possible colors to x in each 2-list assignment.

Thus we can assume that x is really 3-colorable, otherwise there is a natural 2-constraint on it. Since we assume that the color of x is A , we can consider that in all the following 2-list assignments, the constraint $\{B, C\}$ is given to the A -edge-neighborhood of x . Let us abuse notation and still denote by (A, B, C) the partition of the C -edge-neighborhood of x , induced by the solution (A, B, C) . Since there exists a solution where x is colored by C , and C is a AB -clique, then Claim 5.26 ensures

$f(v)$	$f'(v)$	$f''(v)$
A_2 or A_1, A_2	*	C
A_3 or A_1, A_3	*	B, C
A_4 or A_1, A_4	*	A
A_2, A_4	*	A, C
A_2, A_3	*	B, C
A_3, A_4	A'_2 or A'_1, A'_2	B
A_3, A_4	A'_3 or A'_1, A'_3	A, C
A_3, A_4	A'_4 or A'_1, A'_4	C
A_3, A_4	A'_2, A'_4	B, C
A_3, A_4	A'_2, A'_3	A, B
A_3, A_4	A'_3, A'_4	A, C

Figure 5.11: This table describes the rules used in proof of lemma 5.25 to build a 2-list assignment f'' for 3-CCP from a pair (f, f') of 2-list assignment for two instances of the stubborn problem. Symbol * stands for any constraint. For simplicity, we write X, Y (resp. X) instead of $\{X, Y\}$ (resp. $\{X\}$).

that C is a split graph $C' \uplus C''$ with C' a B -clique and C'' a A -clique. The situation is described in Figure 5.12. Let H be the non-colored graph with vertex set the C -edge-neighborhood of x and with edge set the union of B -edges and C -edges (see Figure 5.13). Moreover, let H' be the non-colored graph with vertex set the C -edge-neighborhood of x and with edge set the B -edges (see Figure 5.14). We consider (H, \mathcal{L}_0) and (H', \mathcal{L}_0) as two instances of the stubborn problem, where \mathcal{L}_0 is the trivial list assignment that gives each vertex the constraint $\{A_1, A_2, A_3, A_4\}$.

By assumption, there exists \mathcal{F} (resp. \mathcal{F}') a polynomial 2-list covering for the stubborn problem on (H, \mathcal{L}_0) (resp. (H', \mathcal{L}_0)). We construct \mathcal{F}'' the set of 2-list assignment f'' built from all the pairs $(f, f') \in \mathcal{F} \times \mathcal{F}'$ according to the rules described in Figure 5.11 (intuition for such rules is given in the next paragraph). \mathcal{F}'' aims at being a polynomial 2-list covering for 3-CCP on the C -edge-neighborhood of x .

The following is illustrated on Figure 5.13 and 5.14. Let \mathcal{S} be the partition defined by $A_1 = \emptyset$, $A_2 = C''$, $A_3 = B \cup C'$ and $A_4 = A$. We can check that A_2 is a stable set and A_4 is a clique (the others restrictions are trivially satisfied by A_1 being empty and \mathcal{L}_0 being trivial). In parallel, let \mathcal{S}' be the partition defined by $A'_1 = \emptyset$, $A'_2 = B$, $A'_3 = A \cup C''$ and $A'_4 = C'$. We can also check that A'_2 is a stable set and A'_4 is a clique. Thus \mathcal{S} (resp. \mathcal{S}') is a maximal solution for the stubborn problem on (H, \mathcal{L}_0) (resp. (H', \mathcal{L}_0)) inherited from the solution $(A, B, C = C' \uplus C'')$ for 3-CCP.

Let $f \in \mathcal{F}$ (resp. $f' \in \mathcal{F}'$) be a 2-list assignment compatible with \mathcal{S} (resp. \mathcal{S}'). Then $f'' \in \mathcal{F}''$ built from (f, f') is a 2-list assignment compatible with (A, B, C) .

Doing so for the B -edge-neighborhood of x and pulling everything back together gives a polynomial 2-list covering for 3-CCP on (K_n, f) .

□

Lemma 5.27 ($2 \Rightarrow 3$). *Suppose for every n and every edge-coloring $f : E(K_n) \rightarrow \{A, B, C\}$, there is a polynomial 2-list covering for 3-CCP on (K_n, f) . Then for every graph G , there is a polynomial CS-separator.*

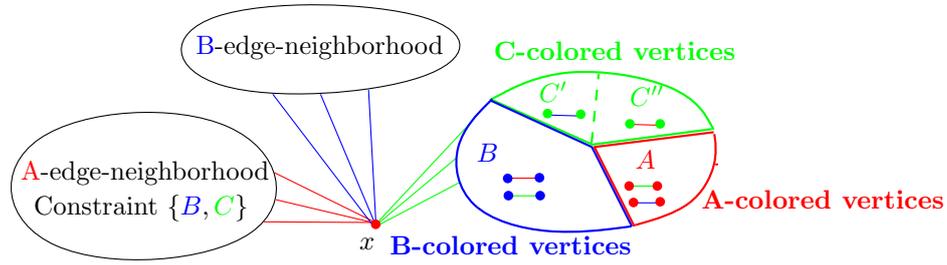


Figure 5.12: Vertex x , its A -edge-neighborhood subject to the constraint $\{B, C\}$, and its C -edge-neighborhood separated in different parts.

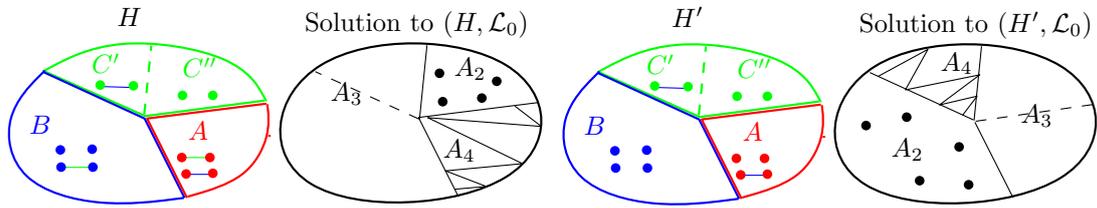


Figure 5.13: On the left the graph H obtained from the C -edge-neighborhood by keeping only B -edges and C -edges. On the right the solution of the stubborn problem.

Figure 5.14: On the left, the graph H' obtained from the C -edge-neighborhood by keeping only B -edges. On the right, the solution of the stubborn problem.

Figure 5.15: Illustration of the proof of lemma 5.25. Color correspondence: A =red ; B =blue ; C =green. As before, cliques are represented by hatched sets, stable sets by dotted sets.

Proof. Let $G = (V, E)$ be a graph on n vertices. Let f be the coloring on K_n defined by $f(e) = A$ if $e \in E$ and $f(e) = B$ otherwise. In the following (K_n, f) is considered as a particular instance of 3-CCP with no C -edge. By hypothesis, there is a polynomial 2-list covering \mathcal{F} for 3-CCP on (K_n, f) . Let us prove that we can derive from \mathcal{F} a polynomial CS-separator \mathcal{C} .

Let $\mathcal{L} \in \mathcal{F}$ be a 2-list assignment. Denote by X (resp. Y, Z) the set of vertices with the constraint $\{A, B\}$ (resp. $\{B, C\}, \{A, C\}$). Since no edge has color C , X is split. Indeed, the vertices of color A form a B -clique and conversely. Given a graph, there is a linear number of decompositions into a split graph [89]. Thus there are a linear number of decomposition $(U_k, V_k)_{k \leq cn}$ of X into a split graph where U_k is a B -clique. For every k , the cut $(U_k \cup Y, V_k \cup Z)$ is added in \mathcal{C} . For each 2-list assignment we add a linear number of cuts, so the size of \mathcal{C} is polynomial.

Let K be a clique and S a stable set of G which do not intersect. The edges of K are colored by A , and those of S are colored by B . Then the coloring $\mathcal{S}(x) = B$ if $x \in K$, $\mathcal{S}(x) = A$ if $x \in S$ and $\mathcal{S}(x) = C$ otherwise is a solution of (K_n, f) . Left-hand side of Figure 5.16 illustrates the situation. There is a 2-list assignment \mathcal{L} in \mathcal{F} which is compatible with this solution. As before, let X (resp. Y, Z) be the set of vertices which have the constraint $\{A, B\}$ (resp. $\{B, C\}, \{A, C\}$). Since the vertices of K are colored B , we have $K \subseteq X \cup Y$ (see right hand-side of Figure 5.16). Likewise, $S \subseteq X \cup Z$. Then $(K \cap X, S \cap X)$ forms a split partition of X . So, by construction, there is a cut $((K \cap X) \cup Y, (S \cap X) \cup Z) \in \mathcal{C}$ which ensures that (K, S) is separated by \mathcal{C} . \square

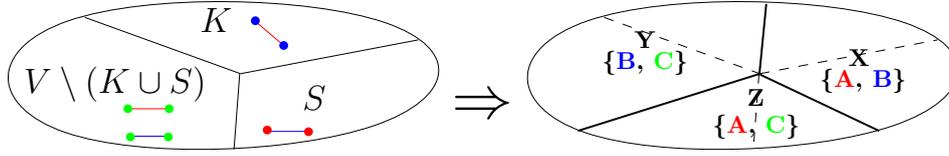


Figure 5.16: Illustration of the proof of Lemma 5.27. On the left hand-side, G is separated in 3 parts: K , S , and the remaining vertices. Each possible configuration of edge- and vertex-coloring are represented. On the right-hand-side, (X, Y, Z) is a 2-list assignment compatible with the solution. X (resp. Y, Z) has constraint $\{A, B\}$ (resp. $\{B, C\}, \{A, C\}$). Color correspondence: **A=red** ; **B=blue** ; **C=green**.

Lemma 5.28 ($3 \Rightarrow 1$). *Suppose for every graph G , there is a polynomial CS-separator. Then for every graph G and every list assignment $\mathcal{L} : V \rightarrow \mathcal{P}(\{A_1, A_2, A_3, A_4\})$, there is a polynomial 2-list covering for the stubborn problem on (G, \mathcal{L}) .*

Proof. Let (G, \mathcal{L}) be an instance of the stubborn problem. By assumption, there is a polynomial CS-separator for G .

Claim 5.29. *If there are p cuts that separate all the cliques from the stable sets, then there are p^2 cuts that separate all the cliques from the unions $S \cup S'$ of two stable sets.*

Proof. Indeed, if (V_1, V_2) separates K from S and (V'_1, V'_2) separates K from S' , then the new cut $(V_1 \cap V'_1, V_2 \cup V'_2)$ satisfies $K \subseteq V_1 \cap V'_1$ and $S \cup S' \subseteq V_2 \cup V'_2$. \square

Let \mathcal{F}_2 be a polynomial family of cuts that separate all the cliques from unions of two stable sets, which exists by Claim 5.29 and hypothesis. Then for all $(U, W) \in \mathcal{F}_2$, we build the following 2-list assignment \mathcal{L}' :

1. If $v \in U$, let $\mathcal{L}'(v) = \{A_3, A_4\}$.
2. If $v \in W$ and $A_3 \in \mathcal{L}(v)$, then let $\mathcal{L}'(v) = \{A_2, A_3\}$.
3. Otherwise, $v \in W$ and $A_3 \notin \mathcal{L}(v)$, let $\mathcal{L}'(v) = \{A_1, A_2\}$.

Now the set \mathcal{F}' of such 2-list assignment \mathcal{L}' is a 2-list covering for the stubborn problem on (G, \mathcal{L}) : let $\mathcal{S} = (A_1, A_2, A_3, A_4)$ be a maximal solution of the stubborn problem on this instance. Then A_4 is a clique and A_1, A_2 are stable sets, so there is a separator $(U, W) \in \mathcal{F}_2$ such that $A_4 \subseteq U$ and $A_1 \cup A_2 \subseteq W$ (see Figure 5.17), and there is a corresponding 2-list assignment $\mathcal{L}' \in \mathcal{F}'$. Consequently, the 2-constraint $\mathcal{L}'(v)$ built from rules 1 and 3 are compatible with \mathcal{S} . Finally, as \mathcal{S} is maximal, there is no $v \in A_1$ such that $A_3 \in \mathcal{L}(v)$: the 2-constraints built from rule 2 are also compatible with \mathcal{S} . \square

Proof of theorem 5.24. Lemmas 5.25, 5.27 and 5.28 conclude the proof of Theorem 5.24. \square

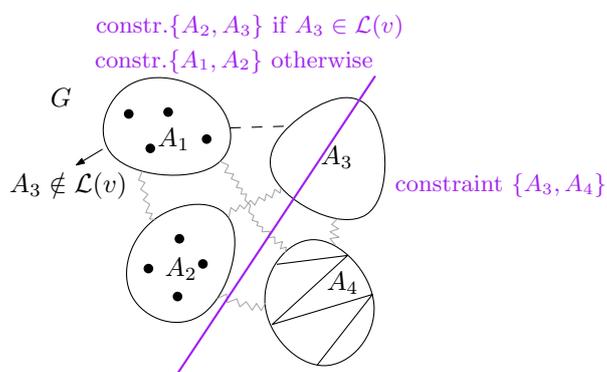


Figure 5.17: Illustration of the proof of Lemma 5.28. A solution to the stubborn problem together with the cut that separates A_4 from $A_1 \cup A_2$. The 2-list assignment built from this cut is indicated on each side.

Multicut is FPT

In this chapter we prove that the MULTICUT problem is FPT parameterized by the solution size. It is joint work with Jean Daligault and Stéphan Thomassé. The proof, and in particular its first part, is based on the important separators technique introduced in Chapter 2. The status of this problem was one of the main open problems in parameterized complexity.

Marx and Razgon independently found a proof that MULTICUT is FPT, with a rather different approach, see [147]. Section 6.5.2 provides a brief comparison between our work and theirs.

Before starting, let us first recall the formal definition of the MULTICUT problem.

MULTICUT:

Input: A graph G , a set of requests R , an integer k .

Parameter: k .

Output: TRUE if there is an (edge)-multicut of size at most k , otherwise FALSE.

6.1 Detailed outline of the proof

Some parts of the proof are technically very involved. This section provides a detailed outline of the proof, underlying the structure of the main results and the reasons behind the main definitions. For formal definitions and statements, and for complete proofs, the reader is referred to the following sections.

A Vertex-Multicut. First of all, we can assume by iterative compression that a vertex-multicut Y of size $k + 1$ is given, and that a solution must split Y . In other words, the solution has to be a MULTIWAY CUT of Y . This is expressed in Lemma 6.9. This vertex-multicut Y gives a first layer of structure to an instance: we can focus on the Y -components, *i.e.* the connected components of the graph where vertices of Y have been removed.

Setting the number of edges of the solution per component. The number of Y -components is bounded in k , considering that all connected components of $G \setminus Y$ which are adjacent to a single given vertex $y \in Y$ form a single Y -component. So, we can branch to decide how many edges of the solution lie in each Y -component.

Half-requests. No request is contained inside a Y -component with two or more attachment vertices, so we can simulate a request (u, v) with several half-requests (u, y, v) , where $y \in Y$ is an attachment vertex of both the component $C(u)$ of u and of the component $C(v)$ of v . Cutting a half-request (u, y, v) means cutting all paths between u and v which go through y .

The goal: 2-SAT. Half-requests give a simpler structure to the multicut problem: cutting a half-request (u, y, v) is equivalent to either separating u from y in $C(u)$, or separating v from y in $C(v)$. We will express this "or" through 2-SAT clauses once we manage to express in a simple way whether the solution separates u from y . The rest of the proof is devoted to simplifying the structure of the instance until we can express with 2-SAT variables whether the solution separates u from y .

Focus on 2-components. We first reduce Y -components with three or more attachment vertices in Lemma 6.12. Now, Y -components have either two attachment vertices (2-components) or one attachment vertex (cherries). The complexity of the problem mostly lies in the existence of 2-components.

In order to give a better structure to 2-components we compute in Lemma 6.14 a particular path between its two attachment vertices: the backbone, which has the following property. The multicut must contain exactly one edge in the backbone. The set of multicuts is thus linearly partially ordered, according to the edge of the backbone they use. The goal is now to simplify the structure of the instance so that the multicuts that separate a vertex u from an attachment vertex y of $C(u)$ form an initial (or final) section of this linear order. Indeed, the fact that the solution belongs to an initial or final section of a linear order can be easily expressed with a 2-SAT variable.

BACKBONE MULTICUT. The instance as reduced up to this point fits the first intermediate variant, COMPONENT MULTICUT, defined in Section 6.2. We introduce in Section 6.3 the second intermediate problem BACKBONE MULTICUT. We need this more general problem than COMPONENT MULTICUT, so that we can enrich instances. Considering half-requests is of major importance in our proof, but the presence of 2-SAT clauses is necessary only for Lemma 6.20, and could possibly be avoided with a slightly different proof.

Lemonizing 2-components. Through Lemma 6.20 in Subsection 6.3.5 we reduce 2-components so that each vertex of the backbone becomes a cut-vertex of the 2-components. An example is drawn in Figure 6.4. This is the first highly technical part of the proof. The components now consist in a sequence of lemons with cherries attached to the backbone.

Linearly ordering the set of multicuts. In Subsection 6.3.6 we perform a complete linearisation of the set of multicuts. This can be seen as the core of our approach. We define a meaningful partial order \leq on multicuts, such that the set of multicuts can be partitioned into a number bounded in k of parts totally ordered by \leq by Dilworth's Theorem. This is Lemma 6.22.

Reducing 2-components to a backbone. The linear order \leq on the set of all multicuts allows us to move terminals of cherries to the backbone in Lemma 6.23. In a component with no cherries left, we manage to reduce the sequence of lemons to just the backbone in Theorem 6.25. This is the second highly technical part of the proof.

Reduction to 2-SAT. At this point, the 2-components are just paths. Only cherries attached to vertices of Y remain, and they have a bounded number of meaningful separators, thanks to the important separators technique. With an instance consisting of a subdivision of a graph with at most k edges, we easily express with 2-SAT variables in Theorem 6.27 whether a given vertex is separated from a given vertex of Y . We end up (after a heavy dose of branching throughout the proof) with a 2-SAT instance, which is polynomially solvable.

Complexity and Programmability. The overall algorithm is single exponential. It should be quite difficult to effectively implement the algorithm though, the whole proof being very involved. Finding a simpler proof leading to a simpler algorithm would be very interesting.

6.1.1 Reductions, branchings and invariants

Let us now turn to the proof of the fixed-parameter tractability of the general MULTICUT problem. In this chapter, we study MULTICUT variants with additional constraints on the deleted edges. In the original MULTICUT problem, we can delete a set of k edges without restrictions, but in some more constrained versions we must delete a prescribed number of edges on some particular paths. The total number of deleted edges is called *deletion allowance* of the multicut problem. We will make extensive use of the term *bounded* which always implicitly means bounded in terms of the deletion allowance. Also, when speaking of FPT time, we always mean $\mathcal{O}(f(d)n^c)$ where c is a fixed constant and d is the deletion allowance. In the algorithm we will perform reductions and branchings, and we use invariants to bound the total running time.

deletion
allowance

Reductions. These are computations where the output is a reduced instance which is equivalent to the original instance with respect to the existence of a solution. One of the most natural reductions concerns irrelevant requests, *i.e.* a request xy such that every k -multicut of $R \setminus xy$ actually cuts x from y , where R is the set of requests. If one can certify that a request xy is irrelevant, the reduction consists in replacing R by $R \setminus xy$. Another reduction is obtained if we can certify that, if there exists a k -multicut, then there exists a k -multicut which does not separate two given vertices u and v . In this case, we simply contract u and v . Reductions are easy to control, and we can perform reductions liberally provided that some invariant polynomial in n decreases. For example, request deletions can be performed at most n^2 times, and vertex contractions at most n times.

Branchings. In our algorithm, we often have to decide if the multicut we are looking for is of a particular type, where the number of types is bounded. We will then say that we branch over all the possible cases. This means that, to compute the result of the current instance, we run our algorithm on each case, in which we force the solution to be of each given type. The instance is positive if at least one of the cases is positive. To illustrate this, in the case of a graph G with two connected components G_1 and G_2 , both containing requests, we would branch over $k-1$ instances, depending of the number of edges (between 1 and $k-1$) that we remove from G_1 . This simple branching explains why we can focus on connected graphs.

Invariants. To prove that the total number of branches is bounded, we show that some invariant is modified at each branching step, and that the number of times that this invariant can be modified is bounded. We usually have several invariants ordered lexicographically. In other words, we have different invariants which we want to increase or decrease and each invariant can take a bounded number of values. These invariants are ordered, there is a primary invariant, a secondary invariant, etc. Each branching must *improve the invariant*, *i.e.* the first invariant (with respect to priority order) which is changed by the branching must be modified according to the preference, increase or decrease, that we specified for it. For instance, the primary invariant could be the number of edges in the multicut, which we want to decrease, and the secondary invariant could be the connectivity of G , which we want to increase. In this case, if we can decrease the number of edges in the solution we do so even if the connectivity of the graph decreases. Also, if a branching increases connectivity and leaves the number of multicut edges unchanged, we improve the invariant.

6.1.2 Irrelevant requests

Using the important separator technique introduced in Chapter 2, let us show that we can assume that every vertex of the graph appears in a bounded number of requests. Section 6.1.2 and 6.1.3 are quite involved and is heavily linked with notions defined in Section 2.3.2. Let $G = (V, E)$ be a graph and x be a vertex call the root. Recall that an important separator S is a subset of vertices containing x such that for every $S' \subsetneq S$, we have $\delta(S) < \delta(S')$. Recall that important separators are closed under union. In addition there are at most 4^k indivisible important xy -separators of size at most k for every vertex v .

We denote by C_k^y , $C_{<k}^y$ and $C_{\leq k}^y$ the set of indivisible important xy -separators of size respectively exactly k , less than k and at most k . We denote by C_k , $C_{<k}$ and $C_{\leq k}$ respectively the union over all vertices y of G of C_k^y , $C_{<k}^y$ and $C_{\leq k}^y$ respectively.

Δ -system

A collection of sets is called a Δ -system if every two distinct sets have the same intersection. Erdős and Rado [83] proved that there exists a function er such that a collection of $er(k, r)$ sets of size at most k contains a Δ -system consisting of r sets. The bound proved in following result will be immediately improved to a single-exponential bound with Theorem 6.2, whose proof is conceptually more complicated.

Theorem 6.1. *Every set K of at least $er(4^k, k')$ vertices contains a subset K' of size k' such that every important separator S with $\delta(S) \leq k$ satisfies either $S \cap K' = \emptyset$ or $|K' \setminus S| \leq k$. In other words, every important separator with border at most k isolates either all the elements of K' , or at most k elements of K' . The set K' can be computed in FPT time in k and k' .*

Proof. Let us consider the collection \mathcal{C} of sets $C_{\leq k}^y$, for all $y \in K$. By Theorem 2.32, the collection \mathcal{C} has size bounded in terms of k and k' and can be computed in FPT time.

The sets $C_{\leq k}^y$ have size at most 4^k and the set K has size $er(4^k, k')$, so there exists a Δ -system of size k' , i.e. a subset K' of k' vertices of K such that for all $y, y' \in K'$, the set $C_{\leq k}^y \cap C_{\leq k}^{y'}$ is equal to some fixed set C of $C_{\leq k}$. This set K' is computable in FPT time. Every indivisible important separator S in C satisfies $S \cap K' = \emptyset$, i.e. the separators in C isolate K' . Moreover, if a separator S in $C_{\leq k}$ does not belong to C , then S belongs to at most one $C_{\leq k}^y$ for $y \in K'$. Indeed otherwise y, y' would be in the same connected component, and then S would be both an indivisible important xy -separator and an indivisible important xy' -separator, so by definition of Δ -system it would be a separator for the whose set K' . So finally S isolates at most one vertex of K' .

We have proved so far that the conclusion of Theorem 6.1 holds if S is an indivisible important separator with border or size at most k , with the stronger conclusion that S isolates at most one vertex of K' when it does not completely separates K' . To conclude, let us observe that if S is divisible and Z is a component of $G \setminus S$, then by Lemma 2.28 the separator \overline{Z} belongs to $C_{\leq k}$. Hence either \overline{Z} isolates K' , or \overline{Z} isolates at most one vertex of K' . The number of components of $G \setminus S$ is at most k , which concludes the proof of Theorem 6.1. \square

The same result holds with a better (single-exponential) bound, as expressed in the following result. We have kept the simpler Theorem 6.1 along with its proof for readers more concerned with the simplicity of the argument than by the efficiency of the algorithm.

Theorem 6.2. *Every set K of at least $\alpha(k, k') = k'^{\binom{k+2}{2}-1}$ vertices contains a subset K' of size k' such that every important separator S with $\delta(S) \leq k$ satisfies either $S \cap K' = \emptyset$ or $|K' \setminus S| \leq k$. The set K' can be computed in FPT (single exponential) time.*

Proof. Observe that the result trivially holds when $k' \leq k$. So we can assume that $k' > k$. We prove the result by induction on k .

For $k = 1$, $\alpha(1, k') = k'^2$. The complements of an important separator with a border of size 1 form a collection of disjoint sets of vertices, which induces a partition of K . There is either a class K' of this partition containing at least $\sqrt{|K|} \geq k'$ elements, or a set K' of size at least $\sqrt{|K|} \geq k'$ whose elements are chosen in different classes. In both cases K' satisfies the induction hypothesis.

Assume now that $k > 1$. We distinguish two cases. Assume that there exists an indivisible important separator S with $\delta(S) \leq k$ and $|K \setminus S| \geq k'^{\binom{k+1}{2}-1}$. By induction, we extract from $K \setminus S$ a subset K' of size k' such that every important separator T with $\delta(T) \leq k-1$ satisfies either $T \cap K' = \emptyset$ or $|K' \setminus T| \leq k-1$. Consider an important separator S' with $\delta(S') = k$. If $S' = S$, then S' isolates K' by definition of K' , hence we assume that S' is distinct from S . Observe that $K' \setminus S'$ is equal to $K' \setminus (S \cup S')$. As $S \cup S'$ is an important separator with border at most $k-1$ (by submodularity we have $\delta(S \cup S') \leq \delta(S) + \delta(S') - \delta(S \cap S')$), the set $K' \setminus S'$ is either K' or has size at most $k-1$. So the conclusion of Theorem 6.2 holds.

Conversely, assume that all indivisible important separators S with $\delta(S) \leq k$ satisfy $|K \setminus S| < k'^{\binom{k+1}{2}-1}$. Consider an auxiliary graph H with vertex set K and where vv' is an edge when there exists an indivisible important separator S with $\delta(S) \leq k$ such that $\{v, v'\} \cap S = \emptyset$. The degree in H of a vertex v is less than $d := k'^{\binom{k+1}{2}-1} \cdot 4^k$. Indeed, there are at most 4^k indivisible important xv -separators with a border of size at most k , and we have assumed that all indivisible important separators S with $\delta(S) \leq k$ satisfy $|K \setminus S| < k'^{\binom{k+1}{2}-1}$. Note that d is less than $k'^{\binom{k+1}{2}-1} \cdot 4^{k'}$ as $k' \geq k$. There is a stable set K' in H of size at least $\frac{|K|}{d}$, and $\frac{|K|}{d} \geq k'$ since $\binom{k+2}{2} - \binom{k+1}{2} - k = 1$ (note that we evaluate 4^k as k^k for simplicity but a better function would work). Every indivisible important separator S with $\delta(S) \leq k$ isolates at most one vertex of K' as K' is stable in H . Thus, every important separator S with $\delta(S) \leq k$ isolates at most k vertices of K' . The induction hypothesis holds in both cases, which concludes the proof of Theorem 6.2. \square

Theorem 6.3. *Every set K with at least $h(\ell) := \ell \cdot 2^{4^\ell} + 1$ vertices of G contains a vertex y such that every separator S with $\delta(S) + |S \cap K| \leq \ell$ is such that $y \notin S$. Moreover, y can be found in FPT time.*

In other words, the vertex y verifies the following: whenever the deletion of a set of a edges isolates x from all but b elements of K , with $a + b \leq \ell$, then the vertex y is also isolated from x .

Proof. Let G' be the graph obtained from G by adding a new vertex z with neighborhood K . In G' , the set C of indivisible important xz -separators with border at most ℓ has size at most 4^ℓ by Theorem 2.32. The size of K is at least $\ell \cdot 2^{4^\ell} + 1$, so there exists a subset T of K of size at least $\ell + 1$ such that for every separator S in C , we have either $T \subseteq S$ or $T \cap S = \emptyset$. Indeed, originally set $T = K$, and for every separator S in C , do $T := T \cap S$ if $|T \cap S| \geq |T \cap \bar{S}|$, and $T := T \cap \bar{S}$ otherwise.

We pick a vertex y in T . Let us prove that y satisfies the conclusion of Theorem 6.3.

In the graph G , consider a set A of a edges which isolates x from all the elements of K apart from a subset B of size b with $a + b \leq \ell$. Let F be the set of edges $A \cup \{zb : b \in B\}$ of G' . Note that F is a

zx -edge separator. We denote by X the component of x in $G' \setminus F$. Since $V(G') \setminus X$ is an indivisible zx -separator with border at most ℓ , it contains by Corollary 2.29 an indivisible important zx -separator U with border at most ℓ . In other words, U belongs to C .

Let us first observe that the set T cannot be disjoint from U . Indeed T has size $\ell + 1$ and each vertex of T is adjacent to z , thus the border of U would exceed ℓ . Hence T is included in U , and the set of edges A isolates T from x in G , and in particular separates y from x . This concludes the proof of Theorem 6.3. \square

This connectivity result allows us to bound the *request degree* of a vertex v , *i.e.* the number of requests with v as an endpoint.

Corollary 6.4. *In a MULTICUT instance with deletion allowance k , the maximum request degree can be reduced to at most $h(k+1) = \mathcal{O}(2^{4^k})$ in FPT time.*

Proof. Consider a vertex x , and denote by K the set of vertices forming a request with x . Assume that $|K| \geq h(k+1)$. In this proof we consider separators rooted in x . By Theorem 6.3, there is a vertex y of K such that every subset S containing x and verifying $\delta(S) + |S \cap K| \leq k+1$ is such that $y \notin S$. We simply remove the request xy from the set of requests. Indeed, let F be a multicut of size at most k of this reduced instance. Let S be the component of x in $G \setminus F$. As F is a multicut, no element of $K \setminus y$ belongs to S (so $|K \cap S| \leq 1$). Moreover $\delta(S) \leq k$ since at most k edges are deleted. Thus $\delta(S) + |S \cap K|$ is at most $k+1$, which implies that $y \notin S$. In other words, even if we do not require to cut x from y , a multicut of the reduced instance must cut the request xy . Therefore removing the request xy from R is correct. When such a reduction can no longer be performed, each vertex has request degree smaller than $h(k+1)$. \square

6.1.3 Cherry reduction

An x -*cherry*, or simply *cherry* is a connected induced subgraph C of G with a particular vertex x called *attachment vertex* of C such that there is no edge from $C \setminus x$ to $G \setminus C$ and no request has its two terminals in $C \setminus x$. In other words, the requests inside an x -cherry must have x as an endpoint. Note that we can always assume that the restriction of a multicut to an x -cherry C is the border of an important separator of C , where x is the root. Indeed it can only be better to put less vertices in the connected component of x . Indeed all the paths between two vertices of a same request must pass through x since no request has both endpoints in the cherry. If $u \in C \setminus x$, a request $uv \in R$ is *irrelevant* if for every multicut F of at most k edges of $R \setminus uv$ and such that $F \cap C$ is the border of an important separator in C , F actually separates u from v .

Theorem 6.5. *Let C be an x -cherry of an instance with deletion allowance k . We can find in FPT time a set $K(C)$ of at most $b(k) := h(k+1)\alpha(k, h(2k+1)) = k^{\mathcal{O}(k^3)}$ terminals in $C \setminus x$, such that if F is a set of at most k edges which separates all requests with one endpoint in $K(C)$ and such that $F \cap C$ is the border of an important separator, then F actually cuts all requests with an endpoint in $C \setminus x$.*

Proof. By Corollary 6.4, we can assume that all terminals have request degree at most $h(k+1)$. Let L be the set of terminals in $C \setminus x$. We assume that $|L| > b(k)$. Our goal is to show that there exists an irrelevant request with one endpoint in L . Let us consider the bipartite request graph B formed by the set of requests with one endpoint in L . The graph B is bipartite since $C \setminus x$ has no internal

cherry

irrelevant

requests. Recall that if a bipartite graph with vertex bipartition (X, Y) has maximum degree d and minimum degree one, there exists a matching with at least $|X|/d$ edges. Indeed, in this case the edges can be partitioned into d matchings and the graph contains at least $|X|$ edges.

The request graph B thus contains a matching M of size at least $\alpha(k, h(2k+1))$ such that each request in M has one endpoint in L and the other endpoint outside $C \setminus x$. Let $K := V(M) \cap V(C \setminus x)$. We first only consider the cherry C where x is the root. Since the size of K is at least $\alpha(k, h(2k+1))$, the set K contains by Theorem 6.1 a subset K' of size $h(2k+1)$ such that every important separator S with border at most k verifies $S \cap K' = \emptyset$ or $|K' \setminus S| \leq k$. Let M' be the set of edges of M having an endpoint in K' . We denote by L' the set of vertices $M' \setminus K'$, i.e. the endpoints of edges in M' which do not belong to $C \setminus x$. Now let us consider the graph $G' := G \setminus (C \setminus x)$ with root x . The set L' has size at least $h(2k+1)$, thus by Theorem 6.3 there is a vertex y in L' such that, whenever we delete k edges in G' such that at most k vertices of L' belong to the component of x , then y does not belong to the component of x . The vertex y being an element of L' , we consider the request $zy \in M'$, where z belongs to $V(C \setminus x)$.

We claim that the request zy is irrelevant. Indeed, let F be a multicut of $R \setminus zy$ with at most k edges such that $F_C = F \cap C$ is the border of an important separator. Let S be the component of x in $C \setminus F_C$. The set S is an important separator and has a border of size at most k , hence, either S completely isolates x from K' or S isolates at most k vertices of K' from x . If K' is isolated from x , then in particular x is disconnected from y , hence the request zy is separator by F . So we assume that a subset K'' containing all but at most k vertices of K' is included in S . Hence, denoting by L'' the other endpoints of the edges of M' intersecting K'' , this means that F must disconnect x from L'' . Therefore, the set F of at most k edges disconnects x from at most $k+1$ elements of L' (the k elements of L'' and possibly y), so by definition of y , the set F disconnects x from y . In particular zy is separator by F . Thus the request zy is indeed irrelevant. All the computations so far are FPT.

We repeat this process, removing irrelevant requests until the size of L does not exceed $b(k)$. We then set $K(C) := L$, and the conclusion of Theorem 6.5 holds. \square

Let C be a cherry of a graph G with deletion allowance k . A subset \mathcal{L} of the edges of C is *active* when we can assume that a multicut uses only edges of \mathcal{L} in C , or more formally: if a multicut F of size at most k exists, then there exists a multicut F' of size at most $|F|$ such that $F' \setminus C = F \setminus C$ and $F' \cap C \subseteq \mathcal{L}$. When the set \mathcal{L} is clear from the context, we say by extension that edges of \mathcal{L} themselves are *active*. active

Lemma 6.6. *Let C be an x -cherry of a graph G with deletion allowance k , and let K be the set of all terminals of $C \setminus x$. Let $\mathcal{L}(C)$ be the union of all borders of separators of $C_{\leq k}^y$, where $y \in K$. Then $\mathcal{L}(C)$ is active, and has size at most $4^{k+1} \cdot |K|$.*

Proof. Assume that F is a multicut of size at most k . Let S be the component of x in $C \setminus F$. Let T be an important separator with $T \subseteq S$ and $\delta(T) \leq \delta(S)$. If a component U of \overline{T} does not intersect K , then $F \setminus \Delta(U)$ is still a multicut. Hence, we can assume that all components U of \overline{T} intersect K , in which case $\Delta(U) \in C_{\leq k}^y$ for some y in K , hence $\Delta(T)$ is included in $\mathcal{L}(C)$. The set $F' = (F \setminus C) \cup \Delta(T)$ is a multicut, and the size bound for $\mathcal{L}(C)$ follows from Theorem 2.32: $C_{\leq k}^y$ contains at most 4^k indivisible important separators of size at most k , and $\mathcal{L}(C)$ is a union of $|K|$ such sets. \square

Theorem 6.7. *Let H_1, H_2, \dots, H_p be x -cherries of a graph G with deletion allowance k such that $H_1 \setminus x, H_2 \setminus x, \dots, H_p \setminus x$ are pairwise disjoint. Assume that for every i , $U_i := H_1 \cup \dots \cup H_i$ is a cherry. Then every set U_i has a bounded active set \mathcal{L}_i such that $\mathcal{L}_j \cap U_i \subseteq \mathcal{L}_i$ whenever $i \leq j$.*

Proof. By Theorem 6.5, we can reduce the set of terminals in U_1 to a bounded set K_1 . The set $\mathcal{L}_1 = \mathcal{L}(U_1)$ is bounded and active by Lemma 6.6. The requests of $C_1 \setminus K_1$ are irrelevant in U_2 since they are irrelevant in U_1 , hence we can assume that Theorem 6.5 applied to U_2 yields a set of terminals $K_2 \subseteq K_1 \cup C_2$. Let \mathcal{L}_2 be the active edges associated to K_2 . Note that if an edge $e \in \mathcal{L}_2$ is in U_1 , then the edge e must belong to a set $C_{\leq k}^y$ for some $y \in K_2 \cap U_1$. Since $K_2 \subseteq K_1 \cup C_2$, we have $y \in K_1$, and so $e \in \mathcal{L}_1$, which is the property we are looking for. We extract K_3 from $K_2 \cup C_3$, and iterate this process to form the sequence \mathcal{L}_i . \square

6.2 Reducing MULTICUT to COMPONENT MULTICUT

Let $G = (V, E)$ be a connected graph, and R be a set of requests. A *vertex-multicut* Y is a subset of V such that every xy -path of G where $xy \in R$ contains a vertex of Y . Let A be a connected component of $G \setminus Y$. We call Y -*component*, or *component*, the union of A and its set of neighbors in Y . Let C be a Y -component, the vertices of $C \cap Y$ are the *attachment vertices* of C .

6.2.1 COMPONENT MULTICUT

Our first intermediate problem is formally expressed below. Informally, the Y -components have at most two attachment vertices. Each Y -component with two attachment vertices has a distinguished path called *backbone*, and the multicut restricted to a component must consist of exactly one edge of the backbone plus a fixed number of other edges. Finally, the vertex-multicut Y must be split by the solution.

COMPONENT MULTICUT:

Input: A connected graph $G = (V, E)$, a vertex-multicut Y , a set of requests, and q integers f_1, \dots, f_q such that:

1. There are q Y -components G_1, \dots, G_q with two attachment vertices x_i and y_i . The other Y -components have only one attachment vertex.
2. Every G_i has an $x_i y_i$ -path denoted P_i called the *backbone* of G_i , such that the deletion of an edge of P_i decreases the edge connectivity in G_i between x_i and y_i .
3. The integers f_1, \dots, f_q are such that $f_1 + \dots + f_q \leq k - q$.

Parameter: k .

Output: TRUE if there exists a multicut F such that:

1. every path P_i contains exactly one edge of F ,
2. every component G_i contains exactly $1 + f_i$ edges of F ,
3. the solution F *splits* Y , *i.e.* each connected component of $G \setminus F$ contains at most one vertex of Y .

Otherwise, the output is FALSE.

free edges

The edges of G which do not belong to the backbones are called *free edges*. The backbone P_i , in which only one edge is deleted, is the crucial structure of G_i . Indeed, the whole proof consists of modifying each Y -component G_i step by step to finally completely reduce it to the backbone P_i . Here, f_i is the number of free edges that we can delete in G_i . Observe that $k - q - f_1 - \dots - f_q$ edges can be deleted in Y -components with one attachment vertex. Our first reduction is the following:

Theorem 6.8. *MULTICUT can be reduced to COMPONENT MULTICUT in FPT time.*

The rest of Section 6.2 is devoted to the proof of Theorem 6.8. We first construct a vertex-multicut Y through iterative compression. Then, we prove that we can reduce to Y -components with one or two attachment vertices. Finally, we show that we can assume that every component with two attachment vertices has a path in which exactly one edge is chosen in the solution. This is our backbone.

6.2.2 The Vertex-Multicut Y

This subsection is devoted to proving by iterative compression that MULTICUT is equivalent to the following problem, as was first noted in [146]:

RESTRICTED MULTICUT:

Input: A graph G , a set of requests R , an integer k , a vertex multicut Y of size at most $k + 1$.

Parameter: k .

Output: TRUE if there is a multicut of size at most k which splits Y , otherwise FALSE.

Lemma 6.9. *MULTICUT is FPT-equivalent to RESTRICTED MULTICUT.*

Lemma 6.9 follows from the following two Lemmas:

Lemma 6.10. *MULTICUT can be solved in time $\mathcal{O}(f(k)n^c)$ if the MULTICUT variant where a vertex multicut of size at most $k + 1$ is additionally given in the input can be solved in time $\mathcal{O}(f(k)n^{c-1})$.*

Proof. By induction on n , we solve MULTICUT in time $f(k)(n - 1)^c$ on $G - v$, where $v \in V(G)$. If the output is FALSE, we return FALSE, otherwise the output is a multicut F of size at most k . Let X be a vertex cover of F of size at most k (i.e. X contains one endpoint of each edge in F). The set $X \cup \{v\}$ is a vertex-multicut of the original instance, so we solve MULTICUT in time $f(k)n^{c-1} + f(k)(n - 1)^c$ which is at most $f(k)n^c$. \square

So we can assume that the input of MULTICUT contains a vertex-multicut Y of size at most $k + 1$.

Lemma 6.11. *We can assume that the solution F splits Y .*

Proof. To a solution F is associated the partition of $G \setminus F$ into connected components. In particular, this induces a partition of Y . We branch over all possible partitions of Y . In a given branch, we simply contract the elements of Y belonging to a same part of the partition. \square

This concludes the proof of Lemma 6.9.

During the following reduction proof, the size of the set Y will never decrease. Since one needs $k + 1$ edges to separate $k + 2$ vertices, the size of Y cannot exceed $k + 1$, otherwise we return FALSE. Hence the primary invariant is the size of Y , and we immediately conclude if we can increase $|Y|$.

6.2.3 Reducing attachment vertices

Our second invariant, which we intend to maximize, is the number of Y -components with at least two attachment vertices. This number cannot exceed k , since a solution must split Y . Our third invariant is the sum of the edge connectivity between all pairs of vertices of Y , which we want to increase. This invariant is bounded by $k \binom{|Y|}{2}$ since the connectivity between two elements of Y is at most k . Note that this third invariant never decreases when we contract vertices.

Lemma 6.12. *If C is a Y -component with at least three attachment vertices, we improve the invariant.*

Proof. Let x, y, z be attachment vertices of C . Let λ be the edge-connectivity between x and y in C . Let P_1, \dots, P_λ be a set of edge-disjoint xy -paths. A *critical edge* is an edge which belongs to some xy -edge separator of size λ . Note that every critical edge belongs to some path P_i . A *slice* of C is a connected component of C minus the critical edges. Given a vertex v of C , the *slice of v* , denoted by $SL(v)$, is the slice of C containing v . Let $B(z)$ be the *border* of $SL(z)$, i.e. the set of vertices of $SL(z)$ which are incident to a critical edge. Note that $B(z)$ intersects every path P_i on at most two vertices, namely the leftmost vertex of P_i belonging to $SL(z)$ and the rightmost vertex of P_i belonging to $SL(z)$. In particular, $B(z)$ has b vertices, where $b \leq 2\lambda$.

We branch over $b + 1$ choices to decide whether one of the b vertices of $B(z)$ belongs to a component of $G \setminus F$ (where F is the solution) which does not contain a vertex of Y . When this is the case, the vertex is added to Y , which increases the primary invariant. In the last branch, all the vertices of $B(z)$ are connected to a vertex of Y in $G \setminus F$. We branch again over all mappings f from $B(z)$ into Y . In each branch, the vertex $v \in B(z)$ is connected to $f(v) \in Y$ in $G \setminus F$. Hence we can contract every vertex $v \in B(z)$ with the vertex $f(v) \in Y$. This gives a new graph G' . We denote by S' the subgraph $SL(z)$ in G' . Observe that S' is a Y -component of G' .

If x and y belong to S' , then the edge connectivity between x and y has increased. Indeed, there is now a path P between x and y inside S' , in particular P has no critical edge. Thus the connectivity between x and y has increased, so the invariant has improved. We assume without loss of generality that x does not belong to S' .

If S' contains an element of Y distinct from z , then S' is a Y -component with at least two attachment vertices. Moreover, there exists a path P in $C \setminus S'$ from x to $B(z)$. Hence we have created an extra Y -component with at least two attachment vertices in G' , which improves the second invariant.

In the last case, z is the only vertex of Y which belongs to S' . Therefore, $B(z)$ is entirely contracted to z . In particular z is now incident to a critical edge e . So there exists an xy -separator A with $\delta(A) = \lambda$ and $e \in \Delta(A)$. Without loss of generality, we assume that $z \notin A$ (otherwise we consider the yx -separator \overline{A}). We denote by B the vertices of \overline{A} with a neighbor in A . In particular, B contains z , has size at most λ , and every xy -path in C contains a vertex of B . Let us denote by L the set $A \cup B$ and by R the set \overline{A} . Note that $L \cap R = B$. We now branch to decide in which components of $G \setminus F$ the elements of B are partitioned. If an element of B is not connected to Y in $G \setminus F$, we improve the invariant. If each element of B is contracted to a vertex of Y , both L and R in the contracted graph are Y -components with at least two attachment vertices (respectively $\{x, z\}$ and $\{y, z\}$). We again improve the invariant. \square

critical edge
slice

6.2.4 Backbones

We now assume that every component has at most two attachment vertices. Let G_1, \dots, G_q be the components of G with two attachment vertices. We denote by λ_i the edge connectivity of G_i between its two attachment vertices x_i and y_i . Recall that the third invariant is the sum of the λ_i for $i = 1, \dots, q$.

Lemma 6.13. *We can assume that x_i and y_i have degree λ_i in G_i .*

Proof. Let A be the unique important $x_i y_i$ -separator with $\delta(A) = \lambda_i$ in the graph G_i rooted in x_i . Let B be the set of vertices of A with a neighbor in \bar{A} . We now branch to decide how the components of $G \setminus F$ partition B . If a vertex of B is not connected to a vertex of Y in $G \setminus F$, we can add it to Y and improve the invariant. If a vertex of B is contracted to a vertex y_i , we increase λ_i . Hence all elements of B are contracted to x_i . Therefore A becomes an x_i -cherry, hence $A \setminus x_i$ is removed from G_i . The degree of x_i inside G_i is now exactly λ_i . We apply the same argument to reduce the degree of y_i to λ_i . \square

We now branch over all partitions of k into $k_0 + k_1 + \dots + k_q = k$, where k_i is the number of edges of the solution chosen in G_i when $i > 0$, and k_0 is the total number of edges chosen in the y -cherries for $y \in Y$.

Lemma 6.14. *Every component G_i can be deleted or has a backbone.*

Proof. If $k_i \geq 2\lambda_i$, then the whole component G_i can be disconnected from the rest of the graph by removing the edges in G_i incident to x_i and y_i . This reduces the second invariant, the number of components with at least two attachment vertices. So we can assume that $k_i \leq 2\lambda_i - 1$. Let $P_1, P_2, \dots, P_{\lambda_i}$ be edge-disjoint $x_i y_i$ -paths.

Our algorithm now branches $2\lambda_i$ times, where the branches are called B_j and B'_j for $j = 1, \dots, \lambda_i$. In the branch B_j , we assume that there is only one edge of the solution selected in P_j , and that this edge is critical, *i.e.* belongs to an $x_i y_i$ -separator of size λ_i . In the branch B'_j , we assume that all the edges of the solution selected in P_j are not critical. Let us show that every solution F belongs to one of these branches. If F does not belong to any branch B'_j , this means that F uses at least one critical edge in each P_j . But since $k_i \leq 2\lambda_i - 1$, some path P_j only intersects F on one edge, which is therefore critical. Hence F is a solution in the branch B_j . Thus this branching process is valid. In the branch B_j , we contract all non critical edges of P_j , therefore P_j is the backbone we are looking for. In the branch B'_j , we contract all critical edges of P_j , hence the connectivity λ_i increases. We thus improve the invariant. \square

This concludes the proof of Theorem 6.8. To sum up, the invariants in this reduction from MULTICUT to COMPONENT MULTICUT are (in decreasing order of importance):

- $|Y|$, to maximize, bounded by $k + 1$.
- The number of components with at least two attachment vertices, to maximize, bounded by k .
- The sum of the connectivity of all pairs of vertices of Y , to maximize, bounded by k^3 .

The tree which represents the branchings made by the algorithm has depth $\mathcal{O}(k^5)$ (the product of the bounds on the invariants). The degree of its nodes is bounded by $\mathcal{O}^*(k^{\mathcal{O}(k)})$. Indeed, when

reducing the components with three or more attachment vertices in Lemma 6.12, or when ensuring that the degree of attachment vertices in a given component is exactly their connectivity in Lemma 6.13, we improve the invariant at a cost of $\mathcal{O}^*(k^{\mathcal{O}(k)})$, and these are the bottlenecks. Finally, the amount of work performed at each node vanishes in front of the number of branches, which gives:

Observation 6.15. *The reduction from MULTICUT to COMPONENT MULTICUT is executed in time $\mathcal{O}^*(k^{\mathcal{O}(k^6)})$.*

Reductions which are only performed once do not impact the overall running time: reducing MULTICUT to RESTRICTED MULTICUT is achieved in time $\mathcal{O}^*(k^{\mathcal{O}(k)})$ in Lemma 6.9; also, there are $\mathcal{O}(k^2)$ components with at most 2 attachment vertices (all y -cherries are considered as a single component for $y \in Y$), hence branching to decide how many edges of the multicut per component costs $k^{\mathcal{O}(k^2)}$. These terms vanish in front of the $\mathcal{O}^*(k^{\mathcal{O}(k^6)})$ term.

6.3 BACKBONE MULTICUT IS FPT

6.3.1 BACKBONE MULTICUT

We introduce here the problem BACKBONE MULTICUT, which is a generalization of COMPONENT MULTICUT. Our goal is to show that BACKBONE MULTICUT is solvable in FPT time, which implies that COMPONENT MULTICUT is FPT, which in turns implies that MULTICUT is FPT thanks to Theorem 6.8.

BACKBONE MULTICUT, formally defined below, differs from COMPONENT MULTICUT in two ways. BACKBONE MULTICUT contains half-requests of the type (u, y, v) , where $u, v \in V$ and $y \in Y$. Cutting the half-request (u, y, v) means cutting all paths from u to v going through y . Also, an instance of BACKBONE MULTICUT can express simple properties on the edge of a backbone selected in the multicut, which allows us to enrich instances.

BACKBONE MULTICUT:

Input: A connected graph $G = (V, E)$, a set R of half-requests, a set Y of at most $k + 1$ vertices, a set B of q variables, a set \mathcal{C} of clauses, and q non negative integers f_1, \dots, f_q such that:

1. G has q Y -components called G_i with two attachment vertices $x_i, y_i \in Y$, where $i = 1, \dots, q$. Moreover, G_i has a backbone Q_i (a prescribed $x_i y_i$ -path) and the $x_i y_i$ -connectivity in G_i is λ_i .
2. The set R contains *half-requests*, i.e. sets of triples (u, y, v) , informally meaning that vertex u sends a request to vertex v via y , where $y \in Y$. Also, Y is a u, v -separator for every half-request $(u, y, v) \in R$.
3. The set B contains q integer-valued variables c_1, \dots, c_q . Each variable c_i corresponds to the deletion of one edge in the backbone Q_i . Formally, if the edges of Q_i are e_1, \dots, e_{ℓ_i} , ordered from x_i to y_i , the variable c_i can take all possible values from 1 to ℓ_i , and $c_i = r$ means that we delete the edge e_r in Q_i .
4. The clauses in \mathcal{C} have four possible types: $(c_i \leq a \Rightarrow c_j \leq b)$, or $(c_i \leq a \Rightarrow c_j \geq b)$, or $(c_i \geq a \Rightarrow c_j \geq b)$, or $(c_i \geq a \Rightarrow c_j \leq b)$.

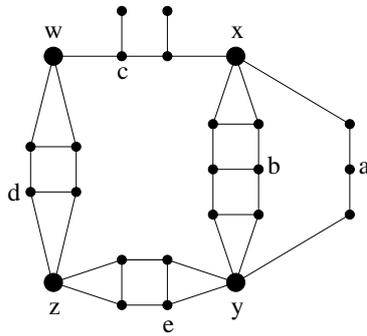


Figure 6.1: Fat vertices w, x, y, z belong to Y . The request ab can be simulated by two half-requests: (a, x, b) and (a, y, b) . The request cd can be simulated by the single half-request (c, w, d) , as paths between c and d which do not go through w have to go through x and y , and hence will be automatically cut when Y is split. By the same argument, the request ce is irrelevant and can be deleted.

5. The integers f_1, \dots, f_q sum to a value at most k . Each integer f_i corresponds to the number of free edges (*i.e.* edges of G which are not in a backbone) of the solution which are chosen in G_i .

Parameter: k .

Output: TRUE if:

1. There exists an assignment of the variables of B which satisfies \mathcal{C} .¹
2. There exists a subset F of at most k free edges of G , which contains f_i free edges in G_i for $i = 1, \dots, q$.
3. The union F' of the set F together with the backbone edges corresponding to the variables of B splits Y and intersects every half-request of R , *i.e.* for every half-request $(u, y, v) \in R$ every path between u and v containing y intersects F' .

Otherwise, the output is FALSE.

Note that the deletion allowance of BACKBONE MULTICUT is $k + q$. COMPONENT MULTICUT directly translates into BACKBONE MULTICUT with an empty set of clauses, and where each request is simulated by one or two half-requests (see Figure 6.1).

This section is devoted to the proof of the following result, which will conclude the proof of the fixed-parameter tractability of MULTICUT:

Theorem 6.16. *BACKBONE MULTICUT can be solved in FPT time.*

6.3.2 Invariants

Our primary invariant is the sum of the numbers of free edges f_i for $i = 1, \dots, q$, which starts with value at most k and is non-negative. A branch in which we can decrease this primary invariant will be considered solved. The secondary invariant is the sum of the $\lambda_i - 1$, called the *free connectivity*,

1. Formally, if c_i is assigned value r , then variables $c_i \leq a$ for $a \geq r$ and variables $c_i \geq a$ for $a \leq r$ are true and variables $c_i \geq a$ for $a \geq r + 1$ and variables $c_i \leq a$ for $a \leq r - 1$ are false.

free connectivity

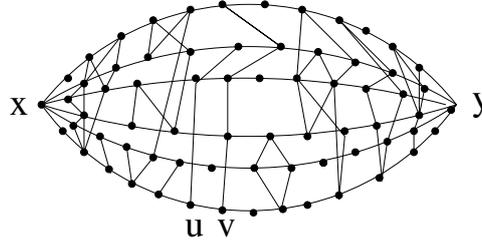


Figure 6.2: A component in a BACKBONE MULTICUT instance with two attachment vertices $x, y \in Y$. The bottom xy -path is the backbone P_1 , and the other "horizontal" xy -paths are the prescribed paths P_2, \dots, P_6 , from bottom to top. Each edge of the backbone does indeed belong to a λ -separator (*i.e.* an xy -separator consisting of six edges). In other words, there exists no path between two distinct vertices of the backbone which consists only of *diagonal edges*, *i.e.* edges which do not belong to the paths P_1, \dots, P_6 . Thus the slices of two distinct backbone vertices are disjoint. The tag of vertex u , as defined in Subsection 6.3.4, is $\{1, 4, 5\}$ and the tag of vertex v is $\{1, 3, 4, 5, 6\}$. The slice connectivity of u is $sc(u) = 6 - 3 = 3$, and $sc(v) = 6 - 5 = 1$. The slice connectivity of this component is 5, as there exists a vertex of the backbone P_1 with no neighbor outside P_1 .

which we try to increase. Observe that this invariant is bounded above by k . For the last invariant, recall that the *slice* $SL(v)$ of a vertex v in a component G_i is the connected component containing v of G_i minus the *critical* edges of G_i , *i.e.* edges of λ_i -separators. Observe that since all edges of a backbone are critical, the slices of distinct vertices in a backbone do not intersect. See Figure 6.2.

The *slice connectivity* of a vertex v in Q_i is the $x_i y_i$ -edge-connectivity of $G_i \setminus SL(v)$ (where $SL(v)$ is considered as a vertex set). We denote it by $sc(v)$. For example, if the set of neighbors of v intersects every $x_i y_i$ -path in $G_i \setminus Q_i$, then we have $sc(v) = 0$. Conversely, if $v \in Q_i$ has only neighbors in Q_i , then $sc(v) = \lambda_i - 1$. The *slice connectivity* sc_i of G_i is the maximum of $sc(v)$, where $v \in Q_i$. The third invariant is the sum sc of the sc_i , for $i = 1, \dots, q$, and we try to minimize this invariant. Observe that sc is always at most k .

Our goal is to show that we can always improve the invariant, or conclude that $\lambda_i = 1$ for all i .

In the following, we consider a component G_i with $\lambda_i > 1$, say G_1 .

To avoid cumbersome indices, we assume that the attachment vertices of G_1 are x and y , and that their edge-connectivity is denoted by λ instead of λ_1 . Moreover, we denote by P_1 the backbone of G_1 , and we assume that $P_1, P_2, \dots, P_\lambda$ is a set of edge-disjoint xy -paths in G_1 . We visualize x to the left and y to the right (see Figure 6.2). Hence when we say that a vertex $u \in P_i$ is to the left of some vertex $v \in P_i$, we mean that u is between x and v on P_i .

6.3.3 Contracting edges

In our proof, we contract edges of the backbone and also free edges which are not critical. We always preserve the fact that the edges of the backbone are critical.

When contracting an edge of the backbone P_1 , we need to modify several parameters. Assume that the edges of P_1 are e_1, \dots, e_ℓ . The variable c_1 represents the edge of P_1 which belongs to the multicut. Now assume that the edge $e_i = v_i v_{i+1}$ is contracted. All the indices of the edges which are at least $i + 1$ are decreased by one. All the constraints associated to the other backbones are not

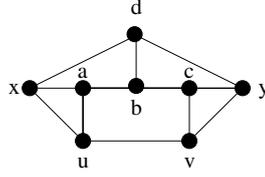


Figure 6.3: This component has two attachment vertices x and y . Its backbone is $P_1 = xuvy$, and we have $P_2 = xabcy$ and $P_3 = xdy$. Both u and v have tag $t(u) = t(v) = \{1, 2\}$, since all edges of P_2 are edges of λ -separator. After contracting u and v , edges ab and bc are no longer edges of λ -separator, and the slice of the resulting vertex $u = v$ becomes $\{u = v, a, b, c, d\}$, and its tag becomes $\{1, 2, 3\}$, which strictly contains $t(u) \cup t(v)$.

affected by the transformation. However, each time a clause contains a literal $c_1 \geq j$, where $j > i$, this literal must be replaced by $c_1 \geq j - 1$. Similarly, each occurrence of $c_1 \leq j'$ for $j' \geq i$ must be replaced by $c_1 \leq j' - 1$. If a set of edges is contracted, we perform the contractions one by one.

The collection of paths P_2, \dots, P_λ can be affected during our contractions since it can happen that a path P_i with $i \geq 2$ contains both endpoints of a contracted edge uv . In such a case, we remove from P_i the loop formed by the contraction, *i.e.* the subpath of P_i between u and v . We thus preserve our path collection.

6.3.4 Choosing a stable edge

Let v be a vertex of P_1 . The *tag* of v is the subset $t(v) := \{i \mid P_i \cap SL(v) \neq \emptyset\}$, *i.e.* the set of indices of the paths intersecting the slice of v . Note that $t(v)$ contains 1. Observe also that the slice connectivity of G_1 is the maximum of $\lambda - |t(v)|$, where v belongs to P_1 . See Figure 6.2.

By extension, the *tag* of an edge $v_i v_{i+1}$ of the backbone P_1 is the ordered pair $(t(v_i), t(v_{i+1}))$. When speaking of an *XY-edge*, we implicitly mean that its tag is (X, Y) . In particular, the edge of P_1 which is selected in the solution has a given tag. We branch over the possible choices for the tag XY of the deleted edge of P_1 . Let us assume that the chosen edge has tag XY .

Lemma 6.17. *If $X \neq Y$, we improve the invariant.*

Proof. Since only one edge is cut in the backbone, we can contract all the edges of P_1 with tags different from XY . Observe that when contracting some UV -edge of P_1 , the tag of the resulting vertex contains $U \cup V$ since the slice of the resulting vertex contains the union of both slices (it can actually be larger, see Figure 6.3). After contraction, all the edges of P_1 between two consecutive occurrences of XY -edges are contracted, hence the tag of every vertex of P_1 now contains $X \cup Y$. In particular, the slice connectivity of G_1 decreases while the free connectivity is unchanged. Thus the invariant has improved. \square

Therefore we may assume that we choose an XX -edge in the solution. Let us contract all the edges of P_1 which are not XX -edges. By doing so, the tag of every vertex of P_1 contains X . After this contraction, the instance is modified, hence we have to branch again over the choice of the tag of the edge chosen in the solution. Any choice different from XX increases the slice connectivity. Hence we can still assume that the tag of the chosen edge is XX .

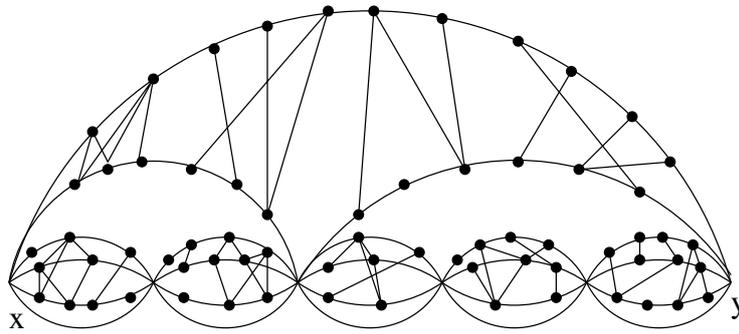


Figure 6.4: All backbone vertices of this component are full, where $X = \{1, 2, 3, 4\}$, with the backbone P_1 at the bottom.

The slice connectivity of G_1 is $\lambda - |X|$. An XX -edge uv of the backbone is *unstable* if, when contracting uv , the tag of the vertex $u = v$ increases (*i.e.* strictly contains X). Otherwise uv is *stable*. We branch on the fact that the chosen XX -edge is stable or unstable.

Lemma 6.18. *If the chosen XX -edge is unstable, we improve the invariant.*

Proof. We enumerate the set of all unstable edges from left to right along P_1 , and partition them according to their index into the odd indices and the even indices. We branch according to the index of the chosen unstable edge. Assume for instance that the chosen unstable edge has odd index. We contract all the edges of P_1 except from the odd unstable edges. We claim that the tag of every vertex now strictly contains X . Indeed, all edges of P_1 between two consecutive odd unstable edges are contracted, in particular some even unstable edge. Thus, since this even edge is unstable, the tag now strictly contains X . Hence the slice connectivity decreases. \square

6.3.5 Contracting slices

In this part, we assume that the chosen edge of P_1 is a stable XX -edge. A vertex v of P_1 is *full* if v belongs to every P_i , where $i \in X$ (see Figure 6.4). Our goal in this subsection is to show that we can reduce to the case where $X = \{1, \dots, \lambda\}$. By the previous section, a branching which increases the tag of the chosen edge would improve the invariant. So we assume that in all our branchings, the chosen edge is still a stable XX -edge.

Lemma 6.19. *We can assume that all backbone vertices are full.*

Proof. We can first assume that there are at most k vertices with tag X between two full vertices. Indeed, let us enumerate w_1, w_2, \dots the vertices with tag X from left to right along the backbone P_1 . A solution F contains at most k free edges and the slices of the vertices of the backbone are disjoint, so at most k slices of vertices w_i contain an edge of F . Hence, if we partition the set of all slices $SL(w_i)$ into $k + 1$ classes according to their index i modulo $k + 1$, the solution F will not intersect one of these classes. We branch on these $k + 1$ choices. Assume for instance that F does not contain an edge in all $SL(w_i)$ where i divides $k + 1$. Therefore, we can safely contract each of such slices $SL(w_i)$ onto w_i . This makes w_i a full vertex.

Let us now enumerate the full vertices z_1, z_2, \dots from left to right. Let uv be a stable XX -edge. There exists a full vertex z_i to the left of u (with possibly $z_i = u$) and a full vertex z_{i+1} to the right of v . Since the number of vertices with tag X between z_i and z_{i+1} is at most k , the number of XX -edges between z_i and z_{i+1} is at most $k + 1$. The *rank* of uv is the index of uv in the enumeration of the edges between z_i and z_{i+1} from left to right. Every edge of P_1 has some rank between 1 and $k + 1$. In particular, we can branch over the rank of the selected stable XX -edge. Assume for instance that the rank of the chosen edge is 1. We then contract all edges which are not stable XX -edges with rank 1. This leaves only full vertices on P_1 since by construction there is a full vertex between two edges of the same rank. \square

Note that after performing the reduction of Lemma 6.19, if $v_i v_{i+1}$ is a stable XX -edge, then for every vertex $w \in P_j$ with $j \in X$ which lies between v_i and v_{i+1} in P_j , every wY -path contains v_i or v_{i+1} . In particular, if $X = \{1, \dots, \lambda\}$ then v_i is an xy -separator-vertex in G_1 , and v_{i+1} as well.

Lemma 6.20. *We can assume that $X = \{1, \dots, \lambda\}$. In other words, we can reduce to the case where every vertex of P_1 is a cut-vertex of G_1 .*

Proof. Assume that X is not equal to $\{1, \dots, \lambda\}$. We show that we can partition the component G_1 into two components G_1^1 and G_1^2 . This partition leaves the free-connectivity unchanged, but decreases the slice connectivity. A vertex v_i of the backbone P_1 is *left clean* if the edge $v_{i-1}v_i$ of P_1 is a stable XX -edge, but the edge $v_i v_{i+1}$ of P_1 is not. The vertex v_i is *right clean* if the edge $v_i v_{i+1}$ is a stable XX -edge, but the edge $v_{i-1}v_i$ is not. Finally, v_i is *clean* if both $v_{i-1}v_i$ and $v_i v_{i+1}$ are stable XX -edges. When enumerating all left clean and right clean vertices from left to right, we obtain the sequence of distinct vertices $r_1, l_1, r_2, l_2, \dots, r_p, l_p$ where the r_i are the right clean vertices and the l_i are the left clean vertices. Observe that x and y do not appear in the sequence since their tag is $\{1, \dots, \lambda\}$. Let us consider a pair r_i, l_i . We say that a vertex v of G_1 is *between* r_i and l_i if every path from v to x or y intersects $\{r_i, l_i\}$. Let B_i be the set of vertices which are between r_i and l_i . Let B be the union of B_i for $i = 1, \dots, p$. left clean

Let G_1^1 be a copy of the subgraph induced by B on G_1 . Observe that G_1^1 has p connected components, since $l_i \neq r_{i+1}$. We contract in G_1^1 the vertices l_i and r_{i+1} , for all $i = 1, \dots, p - 1$, hence making G_1^1 connected. Finally, we contract the vertex r_1 of G_1^1 with x , and we contract the vertex l_p of G_1^1 with y , so that G_1^1 is a Y -component which has x and y as attachment vertices. The backbone P_1^1 of G_1^1 simply consists of the edges of the original backbone.

To construct G_1^2 , we remove from G_1 all the vertices of B which are not left clean or right clean vertices. Hence no stable XX -edge is left in G_1^2 . We contract backbone edges of G_1^2 as follows. All the backbone vertices between x and r_1 are contracted to a vertex $w_1 := x$, more generally all the vertices between l_i and r_{i+1} are contracted to a new vertex called w_{i+1} , and finally all the vertices between l_p and y are contracted to $w_{p+1} := y$. We add in G_1^2 the path w_1, w_2, \dots, w_{p+1} which is the backbone P_1^2 of G_1^2 . We correlate the edges of the backbone of G_1^1 and G_1^2 by adding clauses implying that the chosen edge of P_1^2 is $w_i w_{i+1}$ if and only if the chosen edge of P_1^1 is between r_i and l_i . We finally branch to split the number of free edges f_1 chosen in G_1 into $f_1^1 + f_1^2 = f_1$, the respective free edges deleted in G_1^1 and G_1^2 . Let us call G' the graph G in which G_1 is replaced by G_1^1 and G_1^2 . Note that the free edges of G_1 are partitioned into the free edges of G_1^1 and of G_1^2 . Observe that the free-connectivity of G and G' are equal. However, the slice connectivity has decreased in G' , since its value is 0 in G_1^1 and strictly less than $\lambda - |X|$ in G_1^2 . Indeed, for $i = 1, \dots, p - 1$, either the edge $l_i l_{i+1}$

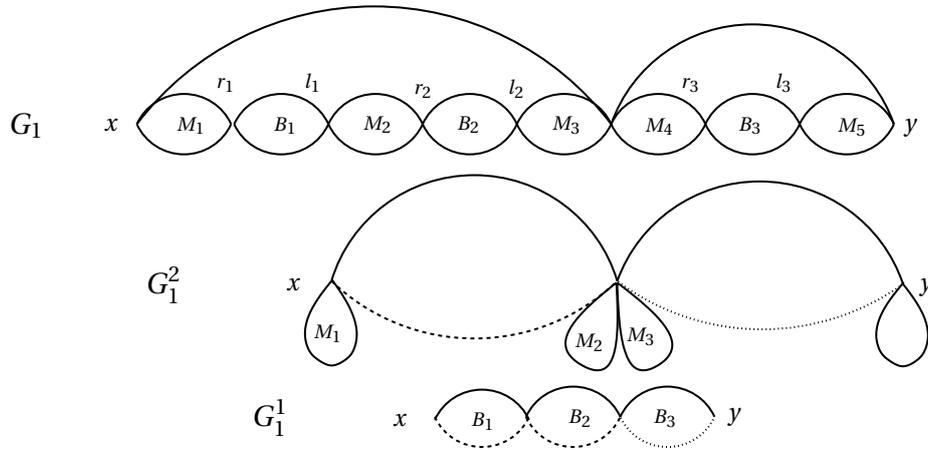


Figure 6.5: At the top, a component G_1 before transformation in the proof of Lemma 6.20. The new components G_1^1 and G_1^2 are depicted at the bottom and in the center respectively. The backbone edge in G_1^2 between w_i and w_{i+1} (dashed, mixed or dotted) is correlated to the subpath of the backbone contained in B_i in G_1^1 (similarly dashed, mixed or dotted). The vertices w_1 and r_1 are identified with vertex x , and the vertices w_4 and l_3 are identified with y , so that the xy -component G_1 is replaced by two xy -components G_1^1 and G_1^2 .²

is unstable or the tag of l_i or l_{i+1} strictly contains X . Hence, contracting all vertices between l_i and r_{i+1} strictly increases the tag of the resulting vertex in G_1^2 . So the invariant improves. Figure 6.5 gives an example of this transformation.

Remains to prove that there exists a multicut in G' if and only if there exists a multicut in G which uses a stable XX -edge. This comes from the following observation. Let $e = v_j v_{j+1}$ be a stable XX -edge of P_1 between r_i and l_i . Let G_e be the graph obtained from G by deleting e , contracting x with all the vertices of P_1 to the left of v_j , and contracting y with all the vertices of P_1 to the right of v_{j+1} . Let G'_e be the graph obtained from G' by deleting e in P_1^1 , deleting the edge $w_i w_{i+1}$ correlated to e in P_1^2 , contracting x with all the vertices of P_1^1 to the left of v_j and all the vertices of P_1^2 to the left of w_i , and contracting y with all the vertices of P_1^1 to the right of v_{j+1} and all the vertices of P_1^2 to the right of w_{i+1} . The key fact is that G_e is equal to G'_e . Hence the multicuts in G and G' selecting the edge e are in one to one correspondence. \square

The proof of Lemma 6.20 produces a new component, hence a new edge to be chosen in a backbone. This increases the deletion allowance by 1, but the number of free edges has not increased. The slice connectivity decreases, so the invariant improves.

2. The names M_i are not used in the proof of Lemma 6.20, their purpose is just to identify what vertices which do not belong to a set B_i become in G_1^2 .

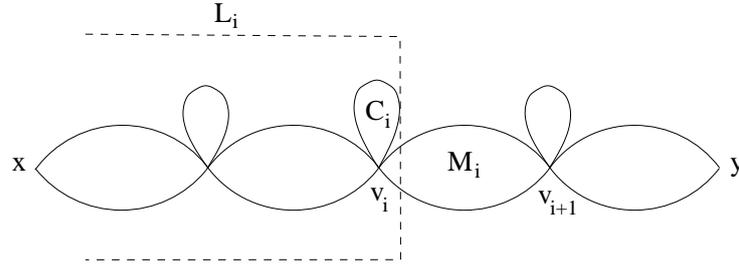


Figure 6.6: The left subgraph L_i of v_i . The set C_i is the v_i -cherry, and M_i is the lemon of the backbone edge $v_i v_{i+1}$.

6.3.6 Reducing the lemons

Thanks to the results of the previous section, we can assume that each vertex of the backbone P_1 of G_1 intersects all paths P_i for $i = 1, \dots, \lambda$. Let $v_i v_{i+1}$ be an edge of the backbone P_1 . The v_i -cherry C_i is the set of all vertices u of G_1 such that every uY -path contains v_i .

The *lemon* M_i of $v_i v_{i+1}$ is the set consisting of v_i , v_{i+1} and of all vertices u of G_1 which do not belong to a cherry and such that every ux -path in G_1 contains v_i and every uy -path in G_1 contains v_{i+1} . Observe that when contracting $v_i v_{i+1}$, the lemon M_i becomes part of the v_i -cherry, where v_i denotes the resulting vertex. We denote by L_i the union of all C_j with $j \leq i$ and all M_j with $j < i$. We call L_i the *left subgraph* of v_i . Similarly, the *right subgraph* R_i of v_i is the union of all C_j with $j \geq i$ and all M_j with $j > i$. See Figure 6.6.

If a multicut F selects the edge $v_i v_{i+1}$ in the backbone, then the vertices x, v_1, \dots, v_i all lie in the same connected component of $G \setminus F$. When these vertices x, v_1, \dots, v_i are contracted to x , the set L_i becomes an x -cherry. Half-requests through y with an endpoint in L_i are automatically cut since F splits Y . Consider the terminals T_i of half-requests of L_i which are routed via x . Note that these half-requests are equivalent to usual requests, since L_i is now an x -cherry. By Theorem 6.5 we can reduce T_i to a bounded set of terminals K_i . This motivates the following key definition.

By Lemma 6.6, we define \mathcal{L}_i to be a bounded active set of edges in the x -cherry obtained from L_i by contracting vertices x, v_1, \dots, v_i . By Theorem 6.7, we can compute such sets \mathcal{L}_i so that $\mathcal{L}_j \cap L_i \subseteq \mathcal{L}_i$ when $i \leq j$.

Let us say that a multicut F selecting $v_i v_{i+1}$ in P_1 is *proper* if $F \cap L_i$ is included in \mathcal{L}_i .

Lemma 6.21. *If there exists a multicut F of size at most k containing the backbone edge $v_i v_{i+1}$, then there is a proper multicut F' of size at most k containing $v_i v_{i+1}$.*

Proof. Consider a multicut F containing $v_i v_{i+1}$. As the set \mathcal{L}_i is active in the cherry obtained by contracting the path x, v_1, \dots, v_i in L_i , there exists a multicut F' of size k such that $F' \setminus L_i = F \setminus L_i$ and $F' \cap L_i \subseteq \mathcal{L}_i$. Hence F' is proper and contains $v_i v_{i+1}$. \square

We denote by \mathcal{L} the set of all subsets F of size at most k contained in some \mathcal{L}_i . We denote by c the maximum size of a set \mathcal{L}_i . Note that c is bounded in terms of k .

Given two sets $F_i \subseteq \mathcal{L}_i$ and $F_j \subseteq \mathcal{L}_j$ with $j \geq i$, let us write $F_i \leq F_j$ when $F_j \cap L_{i+1} \subseteq F_i$. Observe that \leq is a partial order. A subset \mathcal{F} of \mathcal{L} is *correlated* if:

- elements of \mathcal{F} have the same size, and

- \mathcal{F} is a chain for \leq , i.e. for every F_i and F_j in \mathcal{F} , with $F_i \subseteq \mathcal{L}_i$, $F_j \subseteq \mathcal{L}_j$ and $j \geq i$, we have $F_j \cap L_{i+1} \subseteq F_i$.

Lemma 6.22. *There exists a partition $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{k(2c)^k}$ of \mathcal{L} into $k(2c)^k$ correlated sets.*

Proof. Let us prove by induction on $\ell = 0, \dots, k$ that there exists no antichain for \leq in \mathcal{L} consisting of $(2c)^\ell + 1$ sets of size at most l . This clearly holds for $\ell = 0$. Assume that this holds for $\ell - 1$. By contradiction, let $A = \{F_1, F_2, \dots, F_{(2c)^\ell + 1}\}$ be an antichain of sets of size at most ℓ . Let t_i be an integer such that $F_i \subseteq \mathcal{L}_{t_i}$ for $i = 1, \dots, (2c)^\ell + 1$. We assume that the sets F_i are enumerated in such a way that $t_i \leq t_j$ whenever $i \leq j$. The set F_1 is incomparable to all sets F_i with $i > 1$, hence $F_i \cap L_{t_1+1} \not\subseteq F_1$ for all $i > 1$. In particular $F_i \cap L_{t_1+1}$ is non-empty, hence all sets F_i , for $i = 1, \dots, (2c)^\ell + 1$, have an edge in L_{t_1+1} . The sets F_i such that $t_i = t_1$ have an edge in \mathcal{L}_{t_1} by definition. The sets F_i such that $t_i > t_1$ have an edge in \mathcal{L}_{t_1+1} as $\mathcal{L}_{t_i} \cap L_{t_1+1} \subseteq \mathcal{L}_{t_1+1}$, by definition of the sets \mathcal{L}_i . Since the size of $\mathcal{L}_{t_1} \cup \mathcal{L}_{t_1+1}$ is at most $2c$, there exists a subset B of A of size at least $(2c)^{\ell-1} + 1$ of sets F_i sharing a same edge $e \in \mathcal{L}_{t_1} \cup \mathcal{L}_{t_1+1}$. The set $\{F \setminus e \mid F \in B\}$ has size $|B| \geq (2c)^{\ell-1} + 1$ and is an antichain of sets of size at most $\ell - 1$ by definition of \leq . This contradicts the induction hypothesis.

By Dilworth's Theorem, there exists a partition of \mathcal{L} into $(2c)^k$ sets totally ordered by \leq , which can be refined according to the cardinality to obtain a partition into $k(2c)^k$ correlated sets. Such a partition can be found in FPT time. \square

Let us now consider such a partition $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{k(2c)^k}$ of \mathcal{L} into correlated sets. Observe that by Lemma 6.21 we can restrict our search to multicuts of the following type in G_1 :

- A backbone edge $v_i v_{i+1}$.
- Other edges in the lemon M_i , which separate v_i from v_{i+1} in M_i .
- Edges in \mathcal{L}_i .
- Edges in \mathcal{R}_i , which is defined analogously to \mathcal{L}_i , with the roles of vertices x and y reversed.

Lemma 6.23. *We can assume that there are no cherries C_i . Moreover, if a multicut of size at most k exists, there exists one which contains only edges in one lemon M_i .*

Proof. By Lemma 6.21, if there exists a multicut F containing the backbone edge $v_t v_{t+1}$, then there exists a proper multicut F' containing $v_t v_{t+1}$. By definition $F' \cap L_t \subseteq \mathcal{L}$.

We branch over the existence of a proper solution F' such that $F' \cap L_t \in \mathcal{F}_j$ for $j = 1, \dots, k(2c)^k$, where t is the integer such that $v_t v_{t+1} \in F'$. Let us assume that we are in the branch where $F' \cap L_t \in \mathcal{F}_j$. A backbone edge $v_i v_{i+1}$ is in the support of \mathcal{F}_j if there exists some $F_i \in \mathcal{F}_j$ such that $F_i \subseteq \mathcal{L}_i$. When $v_i v_{i+1}$ is in the support we say that lemon M_i is a support lemon. In this case, there actually exists a unique set in \mathcal{F}_j , which we denote by F_i , such that $F_i \subseteq \mathcal{L}_i$, as \mathcal{F}_j is totally ordered under \leq . Let ℓ be the number of edges of the sets in \mathcal{F}_j .

Claim 6.24. *For all $F_a \in \mathcal{F}_j$, if M_i is a support lemon then $F_a \cap M_i = \emptyset$.*

Proof. As \mathcal{L} contains no backbone edge by definition, it is enough to show that u is not disconnected from v_i in $G_1 \setminus F_a$. As M_i is a support lemon, there exists a set $F_i \in \mathcal{F}_j$ such that $F_i \subseteq \mathcal{L}_i$. Consider a set $F_a \in \mathcal{F}_j$ with $F_a \subseteq \mathcal{L}_a$. If $a \leq i$, then $F_a \subseteq \mathcal{L}_a \subseteq \mathcal{L}_i$, so $F_a \cap M_i = \emptyset$. If $a \geq i$, then $F_a \cap L_{i+1} \subseteq F_i \subseteq \mathcal{L}_i$ as \mathcal{F}_j is correlated, hence $F_a \cap M_i = \emptyset$ holds as well. This completes the proof of Claim 6.24. \square

Consider a vertex u such that either u belongs to some cherry C_i or u belongs to a lemon M_i which is not a support lemon. An edge $v_a v_{a+1}$ in the support *affects* a half-request (u, x, v) if $a < i$ or if $i \leq a$ and the unique set $F_a \in \mathcal{F}_j$ such that $F_a \subseteq \mathcal{L}_a$ separates u from x in G_1 . If $v_a v_{a+1}$ does not affect (u, x, v) , then neither does $v_b v_{b+1}$ when $b \geq a$. Indeed when $b \geq a$, $F_b \subseteq \mathcal{L}_b$ and $F_b \in \mathcal{F}_j$, we have that $F_b \cap \mathcal{L}_a \subseteq F_a$.

Let us now modify the instance. If no edge of the support affects a half request (u, x, v) , where either u belongs to some cherry C_i or u belongs to a lemon M_i which is not a support lemon, we remove (u, x, v) from R and add the half-request (x, x, v) . Otherwise we let $v_a v_{a+1}$ be the support edge with a maximal which affects (u, x, v) . We replace (u, x, v) in R by (v_{a+1}, x, v) . We call this process *projecting* the half-request (u, x, v) . After projecting all half-requests via x with an endpoint in a cherry or in a lemon M_i which is not a support lemon, we decrease f_i by ℓ and contract every edge of P_1 which is not in the support of \mathcal{F}_j . Note that, if $v_i v_{i+1}$ is not in the support, then there remains no half-request via x with an endpoint in M_i .

Assume that F' is a solution of this reduced instance which uses an edge $v_a v_{a+1}$ in the support. Let F_a be the element of \mathcal{F}_j such that $F_a \subseteq \mathcal{L}_a$. Then $F' \cup F_a$ is a solution in the original instance. Indeed the half-requests with an endpoint in the support lemons are cut in $F' \cup F_a$ if and only if they are cut by F' , as F_a does not intersect these lemons by Claim 6.24. Also, the half-requests with an endpoint in the lemons which are not support lemons or with an endpoint in the cherries are cut in the reduced instance if and only if they are cut by F_a in the initial instance by construction.

Conversely, assume that F is a proper solution in the original instance which uses the edge $v_a v_{a+1}$ and such that $F \cap \mathcal{L}_a \in \mathcal{F}_j$. In particular, $F_a = F \cap \mathcal{L}_a$, so $F \setminus F_a$ is a solution of the reduced instance. Indeed, all half-requests (u, x, v) cut by F_a in the original instance are affected by $v_a v_{a+1}$, hence they have been projected to (v_i, x, v) with $i \geq a + 1$, and they are cut by $F \setminus F_a$ in the reduced instance.

The reduction, consisting in projecting all half-requests with an endpoint in a cherry or in a lemon which is not a support lemon, improves the invariant unless $\ell = 0$, *i.e.* unless the proper solution of the original instance with backbone edge $v_i v_{i+1}$ does not use any edge in L_i . In this case, all the requests via x of cherry C_j are projected to v_j , for all j . By the same argument, we can assume that no edge in a proper solution is selected to the right of M_i and that the half-requests via y of C_j are projected to v_j . In this case, there remains no terminal in the cherries, so we simply contract the cherries. We are only left with lemons, and we moreover know that if a solution exists, then there exists a solution which uses only edges in a single lemon. This concludes the proof of Lemma 6.23. \square

Theorem 6.25. *We can assume that G_1 only consists of the backbone P_1 .*

Proof. We assume that $\lambda > 1$ and show that we can improve the invariant. Let us consider a backbone edge $v_i v_{i+1}$. We denote by W the multiset of vertices $\{w_2, \dots, w_\lambda\}$ where w_j is the vertex of the slice S_i of v_i in M_i which belongs to the path P_j and has a neighbor in $M_i \setminus S_i$. In other words, w_j is the rightmost vertex of each path P_j in the slice of v_i . These vertices w_j are not necessarily distinct, for instance if v_i has degree λ in M_i , the slice S_i is exactly $\{v_i\}$ hence all w_j are equal to v_i for $j = 2, \dots, \lambda$. We also denote by $Z = \{z_2, \dots, z_\lambda\}$ the multiset of vertices of the slice T_i of v_{i+1} in M_i which belong respectively to the paths P_2, \dots, P_λ and have a neighbor in $M_i \setminus T_i$.

A multicut F induces a partition of $W \cup Z$ according to the components of $G \setminus F$. A vertex of $W \cup Z$ has three possible *types*: it can be in the same component as x after the removal of F , in

the same component as y , or in another component. Observe that, if two vertices a, b of $W \cup Z$ belong to components distinct from the components of x and y in $G \setminus F$, then F is still a multicut after contracting a and b . Hence F induces a partition of W into three parts, each of which can be contracted, and F remains multicut. We now branch over all partitions of $W \cup Z$ into three parts WZ_x, WZ_y, WZ_u , where WZ_x are vertices which are in the same component as x , WZ_y are vertices which are in the same component as y , and WZ_u are vertices of the same type, possibly disconnected from x and y (but not necessarily so). We branch over all possible partitions of W into WZ_x, WZ_y, WZ_u , and contract in each branch WZ_x to v_i , WZ_y to v_{i+1} , and WZ_u (if not empty) is contracted to a single vertex called u_i . In each branch, these contractions are performed simultaneously in all lemons M_i . We denote by G'_1 the resulting component, by M'_i the contracted lemon M_i , and by S'_i the contracted S_i .

If some vertex of W belongs to WZ_y , or if some vertex of Z belongs to WZ_x , or if WZ_u intersects both W and Z , then the xy edge-connectivity increases in G'_1 . Indeed, in all these cases, there exists an xy -path in G'_1 without edges of $\lambda(x, y)$ -separator in G_1 . This improves the invariant, but we cannot directly conclude since the edges of the backbone may not be critical any longer. Indeed, the connectivity between v_i and v_{i+1} in M'_i could be smaller than the connectivity of another lemon M'_j , in which case the backbone edge $v_j v_{j+1}$ is not critical. To get a correct instance of BACKBONE MULTICUT, we simply branch on the connectivity of the lemon M'_i corresponding to the chosen edge $v_i v_{i+1}$. In the branch corresponding to connectivity l , we contract the backbone edges $v_i v_{i+1}$ where M'_i has connectivity distinct from l .

Hence we can assume without loss of generality that W is partitioned into WZ_u and WZ_x , and that $Z = WZ_y$. As we contract WZ_y to v_{i+1} , the vertex v_{i+1} has now degree λ in M'_i , and T_i is a v_{i+1} -cherry. Let us assume that $WZ_u \neq \emptyset$. As we have contracted the vertices of W to v_i and u_i , the set S'_i has exactly two vertices with a neighbor in $M'_i \setminus S'_i$, namely v_i and u_i . Note that the degree of v_i in $M'_i \setminus S'_i$ is exactly the number of vertices w_j chosen in WZ_x (with multiplicity, since WZ_x is a multiset). We denote this degree of v_i in $M'_i \setminus S'_i$ by d . Note that d does not depend on i since we have chosen in every M_i the same subset WZ_x inside $\{w_2, \dots, w_\lambda\}$.

Let λ_S be the $v_i u_i$ edge-connectivity in S'_i . If $\lambda_S > f_1$, then u_i and v_i cannot be separated, so we contract u_i and v_i . We branch in order to assume that λ_S is some fixed value. In the branch corresponding to connectivity λ_S , we contract backbone edges $v_i v_{i+1}$ where S_i has connectivity distinct from λ_S . Let $P'_1, \dots, P'_{\lambda_S}$ be a collection of edge disjoint paths between u_i and v_i in S'_i . We denote by S' the slice of v_i in S'_i , and once again we consider the rightmost vertices $W' = \{w'_1, \dots, w'_{\lambda_S}\}$ of S' in the paths P'_j . We branch over all possible partitions of W' into W'_x, W'_y, W'_u . Once again, if W'_y is not empty, we increase the connectivity between x and y . Observe that W'_u can be contracted to WZ_u , hence to u_i . In particular if W'_u is not empty, we increase the connectivity between v_i and u_i in S'_i . We iterate this process in S'_i until either W'_u is empty in which case v_i has degree λ_S in S'_i , or λ_S exceeds f_1 in which case we contract v_i and u_i .

We apply Lemma 6.23 to G'_1 . Therefore, we can assume that no cherries are left and that if a solution exists, one multicut is contained in some M'_i . Two cases can happen:

If $f_1 \geq d + \lambda_S + \lambda - 1$, and the edge $v_i v_{i+1}$ is chosen in the backbone, then we can assume that the restriction of the multicut to M'_i simply consists of all the edges incident to v_i and v_{i+1} in M'_i . Indeed v_i is incident to $d + \lambda_S$ free edges, and v_{i+1} is incident to $\lambda - 1$ free edges. This is clearly the best solution since it separates all vertices of $M'_i \setminus \{v_i, v_{i+1}\}$ from v_i and v_{i+1} . Therefore, we project

every request (u, x, v) where $u \in M'_i$ to (v_{i+1}, x, v) and project every request (u, y, v) where $u \in M'_i$ to (v_i, y, v) . Finally we reduce f_1 to 0 and we delete all vertices of G'_1 which are not in P_1 .

Assume now that $f_1 < d + \lambda_S + \lambda - 1$. We branch over $2(\lambda - 1)$ choices, where the branches are named B_j and B'_j for all $j = 2, \dots, \lambda$. In the branch B_j , we assume that only one edge of the solution is selected in P_j , and that this edge is critical. In the branch B'_j , we assume that all the edges of the solution selected in P_j are not critical. In the branch B'_j , we contract non critical edges of P_j and improve the invariant. In the branch B_j , we find a new backbone P_j . In this last case, we delete the edges of P_1 and reduce the number of free edges to $f_1 - 1$. We also translate the clauses in terms of edges of the new backbone P_j . Indeed the number of edges in the backbone of G_1 has changed. Clauses of the form $c_1 \leq i$ become $c_1 \leq \epsilon(i)$ where $\epsilon(i)$ denotes the index of the rightmost edge of P_j in the lemon M'_i .

This branching process covers all the cases where $v_i = u_i$ since in this case $f_1 < 2\lambda - 2$ and therefore one path P_j contains only one edge of the multicut. In the case $v_i \neq u_i$, assume that a multicut F is not of a type treated in one of the branches. In other words, F contains at least two edges in each path P_j for $j = 2, \dots, \lambda$, and at least one of them is critical. Then F contains two edges in each of the d paths P_j not containing u_i since F does not respect the branches B_j for $j = 2, \dots, \lambda$. Also, F contains one edge outside S'_i in each path P_j containing u_i since edges in S'_i are not critical and F is not treated in the branches B'_j . Thus F contains at least $2d + (\lambda - d - 1)$ free edges outside S'_i . Hence less than λ_S edges of F lie in S'_i , thus v_i and u_i belong to the same component in $G - F$. This case is covered in another branch in which v_i and u_i are contracted. Hence this branching process is exhaustive, and this completes the proof of Theorem 6.25. \square

6.3.7 Reducing to 2-SAT

We are left with instances in which the Y -components with two attachment vertices only consist of a backbone. We now reduce the last components.

Lemma 6.26. *We can assume that there is no component with one attachment vertex.*

Proof. Let $Y = \{y_1, \dots, y_p\}$ and let k be the number of free edges in the multicut. A vertex $y_i \in Y$ is *safe* if there is no request between two components attached only to y_i . If y_i is not safe then there is a request (u, y_i, v) , with u and y in components attached to y_i , hence y_i must be either disconnected from u or disconnected from v by the solution. We explore one branch where u is added to Y , and one branch where v is added to Y . This creates a component with two attachment vertices. This component has a backbone, and the number of free edges decreases.

Hence, we can assume that all the vertices of Y are safe. The y_i -cherry is the union of all the components attached to y_i . We branch over all possible integer partitions of k into a sum $k_1 + k_2 + \dots + k_p = k$. In each branch, we require that k_i edges are deleted in the y_i -cherry for $i = 1, \dots, p$. By Lemma 6.6, the y_i -cherry has a bounded active set \mathcal{L}_i , hence in the y_i -cherry we can consider only a bounded number of separators of size k_i : all subsets of \mathcal{L}_i of size k_i . We branch over these different choices. In a given branch, we delete a particular set of edges F_i in the y_i -cherry. Thus, we delete the vertices of the y_i -cherry isolated from y_i by F_i , and contract the other vertices of the y_i -cherry to y_i . Finally, no Y -cherry remains. \square

Theorem 6.27. *Multicut is FPT.*

Proof. By Lemma 6.26, to prove that BACKBONE MULTICUT is FPT, we only have to deal with an input graph G which is a subdivision of a graph with at most k edges, and where a multicut must consist of exactly one edge in each subdivided path. Let us consider a half-request (v_i, x, v'_j) . Assume without loss of generality that $v_i \in G_1$, $v'_j \in G_2$, and x belongs to G_1 and G_2 (if x does not belong to G_1 or G_2 , then splitting Y automatically results in cutting the half-request (v_i, x, v'_j)). For simplicity, we assume that the edges of both P_1 and P_2 are enumerated in increasing order from x . We add to \mathcal{C} the clauses $x_1 \geq i \Rightarrow x_2 \leq j - 1$ and $x_2 \geq j \Rightarrow x_1 \leq i - 1$. We transform all the half requests in this way. We are only left with a set of clauses which we have to satisfy.

We finally add all the relations $x_i \geq a \Rightarrow x_i \geq a - 1$ and $x_i \leq a \Rightarrow x_i \leq a + 1$ and $x_i \geq a \Rightarrow \neg(x_i \leq a - 1)$ and $x_i \leq a \Rightarrow \neg(x_i \geq a + 1)$ to ensure the coherence of a satisfying assignment. We now have a 2-SAT instance which is equivalent to the original multicut instance. As 2-SAT is solvable in polynomial time, this shows that BACKBONE MULTICUT is FPT. Hence the simpler COMPONENT MULTICUT problem is FPT. Together with Theorem 6.8 which reduces MULTICUT to COMPONENT MULTICUT, this concludes the proof of Theorem 6.27. \square

To sum up, the invariants in the proof that BACKBONE MULTICUT is FPT are (in decreasing order of importance):

- The total number of free edges in the multicut, to minimize, bounded by k .
- The sum of the free connectivity in each component, to maximize, bounded by k .
- The sum of the slice connectivity in each component, to minimize, bounded by k .³

The algorithm starts by branching over the tag XY of the backbone edge chosen in the multicut in a given component. When $X \neq Y$, the invariant improves by Lemma 6.17. The dominant complexity term comes from the $\mathcal{O}(2^k)$ branches where $X = Y$. Tags may change, and another such branching is done, and again the dominant term comes from the 2^{2k} cases where $X = Y$. If the chosen edge is unstable, we improve the invariant with a factor two branching in Lemma 6.18. If the edge is stable, we branch over k choices to decide which part of a partition modulo $k + 1$ does not intersect the solution F in Lemma 6.19, and branch again over $k + 1$ choices for the rank of the backbone edge chosen in the solution. This yields $\mathcal{O}(k^2 2^{2k})$ cases where all vertices are full. If a backbone vertex is not a cut-vertex, we increase the invariant by Lemma 6.20. Otherwise, we apply Theorem 6.25, which branches over $3^{\mathcal{O}(k)}$ cases, in which either the component consists only in its backbone, or the invariant has improved.

When the whole process described in the previous paragraph has been performed over all components with two attachment vertices, yielding $\mathcal{O}^*(k^{\mathcal{O}(k)})$ branches, we apply Lemma 6.26. If a vertex in Y is not safe, the invariant improves. When all vertices are safe, the tree which represents the branchings made by the algorithm thus far (where child nodes are instances with a better invariant) has depth at most k^3 and the degree of its nodes is bounded by $\mathcal{O}^*(k^{\mathcal{O}(k)})$. Hence the total number of leaves in the branching process thus far is $\mathcal{O}^*(k^{\mathcal{O}(k^4)})$.

Lemma 6.26 proceeds by branching over $\mathcal{O}(k^k)$ cases to fix the number of edges chosen by the solution in each component, and then applies Lemma 6.6, branching exhaustively over all subsets with the adequate number of active edges in each component. This gives a branching factor of $\mathcal{O}((|K| \cdot 4^k)^k)$, where K is the set of terminals in a given cherry, which is bounded by $\mathcal{O}(k^{\mathcal{O}(k^3)})$ by

3. By "bounded" we mean bounded above, all the invariants described in this chapter being trivially non-negative. In both cases, maximize or minimize, the upper bound corresponds to the maximum number of times an invariant can be improved.

Theorem 6.5. The total number of branches obtained thus far is $\mathcal{O}(k^{\mathcal{O}(k^4)})$. Finally, Theorem 6.27 directly translates to a 2-SAT instance. Thus:

Observation 6.28. *The FPT algorithm for BACKBONE MULTICUT runs in time $\mathcal{O}^*(k^{\mathcal{O}(k^4)})$.*

6.4 Hints for VERTEX MULTICUT

This section contains a sketch of a translation of our proof for edge-multicut in terms of vertex-multicut. The proof has the same outline, and we just explain how the notions introduced for edge-multicut can be transferred to the vertex-multicut setting. What follows is more intended as a hint rather than a complete proof of the fact that the following version of Multicut is FPT:

VERTEX MULTICUT:

Input: A graph G , a set of requests R , a subset of vertices S , an integer k .

Parameter: k .

Output: TRUE if there is a vertex-multicut of size at most k which does not intersect S , otherwise FALSE.

In the standard version of VERTEX MULTICUT, the set S is empty. We use this slightly more general version for technical reasons. Let us now explain how we can translate the results of the previous sections for VERTEX MULTICUT.

The results on important separators are based on the submodularity of edge separators. The vertex separators being also submodular, we can transfer the results for vertices. Here an indivisible xy -separator is a set of vertices K which deletion separates x from y and such that no strict subset of K separates x from y . For the reduction from VERTEX MULTICUT to COMPONENT MULTICUT, the proof is essentially the same. One particularity of VERTEX MULTICUT is the following. When we contract vertices, we have to add the resulting vertex to S , the set of non-deletable vertices. Let Y be the vertex-multicut of size $k+1$ given by iterative compression. We can branch to decide which vertices of Y belong to the solution and then branch over the possible contractions of the set Y . Hence we can assume that $Y \subseteq S$. We have to replace arguments of the type “we add a vertex to Y ” by “we branch to know if the vertex is added to Y or if it belongs to the solution”. The connectivity between x and y is the maximum number of paths between x and y whose intersections with the set of deletable vertices are pairwise disjoint. The connectivity can be computed by flows with weight 1 for deletable vertices and ∞ for non-deletable vertices. A vertex of λ -separator is a deletable vertex which deletion decreases the connectivity. In the vertex-multicut context, a *backbone* is a path in which only one vertex will be selected in the multicut and where every odd vertex belongs to the set S . Additionally, all the deletable vertices of the backbone must be vertices of λ -separator.

To prove the existence of a backbone, we have to generalize Lemma 6.13. The border of the slice of x_i has size at most k but the number of vertices which touch this border can be arbitrarily large. We can branch to know if a vertex is deleted in the slice. If this is not the case then the slice can be contracted to x_i , hence x_i has only λ neighbors. Otherwise, we can branch to know if each vertex in the border is in the component of y_i or a new component. In each of these case the invariant improves. Hence, the only relevant case is when all the vertices of the border of x_i will lie in the same connected component as x_i . In this case, we can contract x_i with the border of its slice, which

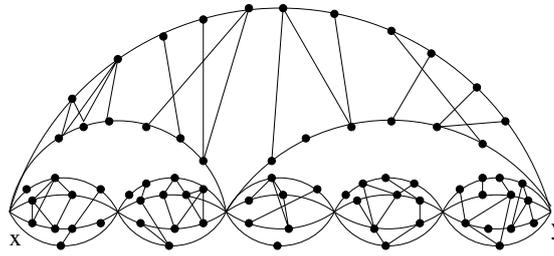


Figure 6.7: The transformation into lemons for VERTEX MULTICUT. The backbone is the path at the bottom. All even vertices of the backbone are full, with $X = \{1, 2, 3, 4\}$, and all odd vertices belong to S .

yields a cherry in which we have to delete vertices. By Lemma 6.6, we can bound the number of possible separators. We can branch these separators and decrease the deletion allowance.

Let us now turn to the BACKBONE MULTICUT problem for vertex-multicuts. A key definition of Section 6.3 is the notion of full vertex. We have to modify this notion, as contracted vertices are not deletable. Hence all the vertices of the backbone cannot be full as for edge-multicuts. Instead, we transform the instance to make all non-deletable vertices of the backbone full (see Figure 6.7). The slice $S(v)$ of a non-deletable vertex v is the connected component of v in G minus the vertices of λ -separator. We define the tag as for edge-multicut. A vertex v of the backbone is X -stable if v is deletable, and the tag of each of its two neighbors in the backbone is X , and the tag after the contraction of v with its two neighbors is still X . As for edge-multicuts, we can assume that we delete an X -stable vertex in the backbone. We can similarly define classes for Lemma 6.19, and remark that one class does not intersect the solution. All the vertices of each slice in this class can be contracted. This ensures that all the vertices which are non-deletable are full. We can write as in Lemma 6.20 that a non-deletable vertex is *left* (resp. *right*) *clean* if the vertex to its left (resp. right) is X -stable and then we can assume that $X = \{1, \dots, \lambda\}$ as for edge-multicut.

In the reduction of the lemons for Vertex-Multicut, we cannot contract x with the border of its slice since it does not ensure that the degree of x is λ . Hence we have to contract x with vertices of its slice which touch the vertices of the border. The set of such vertices can be restricted to a bounded size with Lemma 6.6. Hence the same inductive method used for edge-multicut also holds.

6.5 Conclusion

6.5.1 A single-exponential algorithm

Our proof was originally designed only to prove that MULTICUT is FPT, with no particular focus on algorithmic efficiency. For the sake of completeness, we briefly analyzed the complexity after each of the two major parts of the proof. MULTICUT is reduced to COMPONENT MULTICUT in time $\mathcal{O}^*(k^{\mathcal{O}(k^6)})$ by Remark 6.15, and BACKBONE MULTICUT is solved in time $\mathcal{O}^*(k^{\mathcal{O}(k^4)})$ by Remark 6.28. Hence the overall running time of the algorithm is $\mathcal{O}^*(k^{\mathcal{O}(k^6)})$. This algorithm and its complexity analysis are definitely not fine tuned, and the running time could probably be vastly improved with slight changes to the proof and to the analysis.

6.5.2 Comparison with Marx and Razgon's Proof in [147]

Marx and Razgon independently proved that MULTICUT is FPT in [147]. Both proofs start with the Iterative Compression technique originally used in [146]. Reducing Y -components (our vertex-multicut Y is denoted by W in [147]) with three or more attachment vertices as we do in Lemma 6.12 essentially corresponds to Lemma 5.3 in [147]. The basic connectivity tools, important separators in [147] are identical. At this point, the two proofs drastically diverge. While we concentrate on linearly structuring the (restriction inside a component of the) multicuts, Marx and Razgon focus on *non-isolating solutions*, *i.e.* solutions where no vertex is disconnected from Y . They exhibit a transformation from a positive instance to an instance which admits a non-isolating solution with a large enough probability ($2^{-\mathcal{O}(k^3)}$). This probabilistic transformation can be derandomized into a single-exponential algorithm running in time $\mathcal{O}^*(2^{\mathcal{O}(k^3)})$ (Lemma 4.1). Finally, with an instance admitting a non-isolating solution, they reduce to Almost 2-SAT, which was proved FPT in [171].

Roughly speaking, the two proofs are about as much technically intricate. On the plus side, our proof is self-contained, while [147] uses Almost 2-SAT, which Fixed-Parameter Tractability had remained an important open question until recently. Also, not going through randomization and through Almost 2-SAT allows us to (arguably) get more insight on the structure of the problem. On the minus side, Marx and Razgon's algorithm is more efficient and "cleaner". Also, their proof works directly with vertex-multicuts, while our proof has been originally written in terms of edge-multicut. Finding a shorter proof retaining the best characteristics of each would be very interesting.

6.5.3 Other leads

Considering finer concepts than the notion of request can also be envisioned. In the simpler case of MULTICUT IN TREES, a request is simply a path. In general graphs, a request can be seen as the set of paths between its endpoints. In our proof, we simulate requests by half-requests, partitioning the set of paths naturally associated to a request according to an intermediate point. This could be done thanks to the vertex-multicut Y . But we originally wanted to go much further, and consider the more general problem of cutting a prescribed set of paths, not necessarily all paths between given pairs of vertices. The obvious problem is that a request can be realized by exponentially many paths (exponentially many in n), but we loosely conjectured that this difficulty can be avoided as follows:

Problem 8. *Given a graph G on n vertices, an integer k and two vertices u, v of G , does there always exist a set S of at most $f(k) \cdot \text{Poly}(n)$ paths between u and v such that removing at most k edges of G to disconnect all paths in S must actually disconnect u and v ? In other words, can an FPT number of paths simulate a request with respect to k -multicuts?*

Consider for starters the multigraph G consisting of a path on n vertices with endpoints $u = p_1$ and $v = p_n$, where each edge has been duplicated (into a 2-cycle), as in Figure 6.8. We are looking for a small set of uv -paths, such that every hitting set of these paths is a uv -edge-separator.

Given a (simple) uv -path P and an integer $i \in \{1, \dots, n-1\}$, let us say that P *takes* i if P contains the top edge in *position* i , *i.e.* between p_i and p_{i+1} . We can reformulate the constraint on the solution set S as follows: for every set T of k positions and for every bipartition of T into \overline{T} and \underline{T} ,

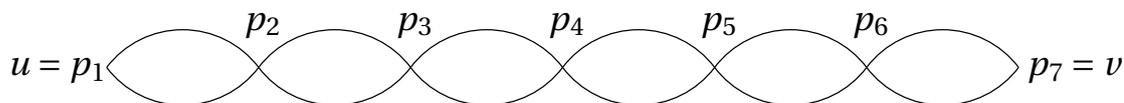


Figure 6.8: A double path on seven vertices.

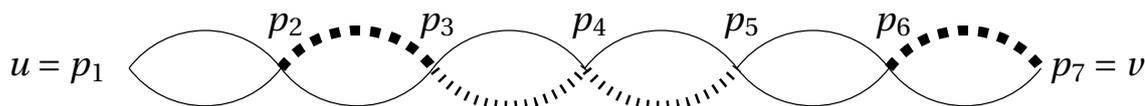


Figure 6.9: Assuming $k = 4$, one constraint would be generated by $T = \{2, 3, 4, 6\}$ partitioned into $\overline{T} = \{2, 6\}$ (corresponding to the dashed edges) in and $\underline{T} = \{3, 4\}$ (corresponding to the dotted edges). A solution must contain a uv -path avoiding all dotted edges and dashed edges.

there must exist a path in S which takes all positions in \overline{T} and takes no position in \underline{T} . Indeed, if this is not the case, then the bottom edges for positions in \overline{T} and the top edges for positions in \underline{T} form a set of k edges which hits S but does not cut u from v . See Figure 6.9 for an example.

Hence every bipartitioned set of k positions gives a constraint. There are $\binom{n}{k} 2^k$ such constraints and a given simple uv -path satisfies (in the above sense) $\binom{n}{k}$ such constraints. Hence a random simple path satisfies a fraction $\frac{1}{2^k}$ of the constraints. In particular, there exists one path P which satisfies at least a fraction $\frac{1}{2^k}$ of the constraints. We pick the path P in our solution S and repeat. This process (not taking into account the computability of P) finds a solution of size at most $\log_{2^k}(\binom{n}{k} 2^k)$ paths, which is actually even better than needed, with a logarithmic dependence on n rather than a polynomial dependence.

Problem 9. *If Conjecture 8 has a positive answer, can such an FPT set of paths emulating the request uv be computed in FPT time?*

In the simple example worked out above, the randomized process should easily be derandomizable into an FPT algorithm.

If Conjecture 8 and Conjecture 9 turn out to have positive answers, then MULTICUT can be reduced to the following problem, already raised in Chapter 2:

HITTING PATH:

Input: A graph G , a set R of paths in G , an integer k .

Parameter: k .

Output: TRUE if there is a set at most k edges of G which hits R , otherwise FALSE.

It is not clear a priori whether this problem should be easier or harder than MULTICUT. Directly emulating a MULTICUT instance with HITTING PATH would require an exponential number of paths, and conversely the structure of the objects to be hit can be more complicated in HITTING PATH than in MULTICUT.

We did not pursue this insight further when the ideas exposed in this chapter proved to be fruitful, but Problems 8, 9 and 2 remain very interesting nonetheless, on their own right as well as with respect to MULTICUT.

Conclusion

Combinatorial bounds on hitting sets using VC-dimension. All along Chapter 3, we gave several upper bounds on the transversality of hypergraphs. They are functions of the fractional transversality, of the VC-dimension, of the (p, q) -property and of the dual 2VC-dimension. Let us summarize all of them and focus on their tightness.

1. Theorem 3.22, due to Haussler and Welzl, ensures that $\tau = \mathcal{O}(d\tau^* \log \tau^*)$ (where d denotes the VC-dimension).
2. Theorem 3.32, due to Ding, Seymour and Winkler, ensures that $\tau = \mathcal{O}(v^{2d+1})$ (where d denotes the dual 2VC-dimension).
3. Theorem 3.36, due to Matoušek, ensures that $\tau \leq f(p, d)$ if the hypergraph has dual VC-dimension d and the $(p, d - 1)$ -property holds.

We have seen that the bound of Theorem 3.22 is tight. Closing the gap between the upper and the lower bounds for the two other theorems is a challenging open problem. For the last two theorems, there is (up to my knowledge) no construction which provides non-trivial lower bounds. Finding such a construction is an interesting problem.

Improving the upper bounds of the last two theorems is even more challenging. Indeed it would also improve the bounds of every theorem with a proof based on these results. For instance, the results of Section 3.4.2 and Chapter 4 are based on respectively Theorem 3.32 and Theorem 3.36. Let us now detail for each of them the prospects of improvements.

- The proof of Theorem 3.32 consists in proving two distinct inequalities. More precisely they first proved that $\tau \leq f(\tau^*)$ using the VC-dimension and then they showed that $v^* \leq g(v)$. Even if both of them can be tight (which is not clear for the second inequality), it would be really surprising that they are tight for the same hypergraphs. Hence the bound may be improved and if we prove both inequalities “at the same time”. Note nevertheless that the current proof is already tricky, so improving it would not be simple.
- For Theorem 3.36, the function f is not specified in the paper of Matoušek. The proof of Matoušek is derived from the proof of the Hadwiger-Debrunner Conjecture, due to Alon and Kleitman. There exist more recent proofs of this result. Maybe some of them can be adapted

for the VC-dimension in order to improve (and explicit) the function f . An ambitious but interesting question would be to determine if τ can be polynomially bounded by a function of p and d .

Result presented in the manuscript. Let us recall in this paragraph the results obtained during my PhD and are presented throughout this manuscript.

- In Chapter 3, we proved that the Scott’s conjecture holds for maximum triangle-free graphs using Theorem 3.32. More precisely we proved that every maximal triangle-free graph with no induced subdivision of a fixed graph H has chromatic number at most $\mathcal{O}(e^{c \cdot \ell^4})$ where c is a constant. The upper bound is probably not tight and it could be interesting to look for a better upper bound, in particular a polynomial one.
- In Chapter 4 we generalized a result due to Chepoi, Estellon and Vaxès on planar graphs for graphs of “bounded VC-dimension”. Recall that we defined the VC-dimension of a graph as the maximum over all induced subgraphs of the VC-dimension of the B -hypergraph (hyperedges are balls of all possible centers with all possible radii). We first proved that clique-minor free graphs as bounded rankwidth graphs have bounded 2VC-dimension. We finally extended the result of Chepoi, Estellon and Vaxès to graphs of bounded 2VC-dimension (using Theorem 3.32) or of bounded VC-dimension (using Theorem 3.36).
- In Chapter 5 we studied the Yannakakis’ CL-IS conjecture and the Alon-Saks-Seymour conjecture. We proved that these two conjectures are equivalent. An implication of this equivalence was known before our results. We also proved that the Yannakakis conjecture holds for several classes of graphs: random graphs, induced split-free graphs (the proof is based on Theorem 3.22) but also (P_k, \overline{P}_k) -induced free graphs, whose proof is a consequence of another result obtained during my PhD on the Erdős-Hajnal conjecture.
- Finally we studied MULTICUT from a parameterized point of view in Chapter 6. The proof is based on the extensive use of the important separators technique introduced by Marx for the study of graph separation problems.

Further applications of VC-dimension. Throughout this manuscript, we raised several open problems. In the next few paragraphs we will focus on some of them which may be tackled with tools presented in this manuscript. The following Gyarfás’ conjecture was raised in Section 3.3.2: any tournament which is the disjoint union of k partial orders admits a dominating set of size at most $f(k)$ (see Section 3.3.2 for more details). This conjecture is open even for $k = 3$. The VC-dimension seems to be a natural tool for studying this conjecture. Even though, the VC-dimension of the neighborhood hypergraph of a tournament which is the disjoint union of 3 orders can be arbitrarily, it does not mean that the method is useless. Indeed, a lot of structure is enforced in the graph induced by a shattered set. So maybe one can hope to prove Gyarfás’ conjecture (at least for $k = 3$) using the following method:

- Partition the graph into two parts. The first one contains all the vertices which appear in at least one large shattered set. The other contains the remaining vertices.
- The second part of the partition has a bounded dominating set. Indeed the scheme of the proof of Alon et al. for k -majority tournaments presented in Chapter 3 can be adapted.
- The first part contains vertices which are in large shattered sets. So the set has some structural

properties: it may be enough for proving that the dominating set is bounded. In particular one can prove that if there is only one large shattered set, then the dominating set equals two. Though, note that this part is the complex part and it is not clear that these structural properties are enough.

HITTING PATH and shadow removal. In Chapter 2, we have also raised the following question: given a set of paths and an integer k , is it possible to decide if the set of paths has a hitting set of size at most k in FPT time? This question naturally generalizes the MULTICUT problem. The shadow removal technique, presented in Chapter 2, can help us to find some structure on the solution we are looking for. Indeed after applying the main theorem of the shadow removal technique, we can assume that the solution has no shadow. This information on the structure of the solution can be interesting to design an FPT algorithm. Indeed in this case, we might find an edge which can be contracted in polynomial time (or any other decreasing invariant).

Another challenging open problem on important separators (and then on shadow removal technique) is a generalization of these tools for separators satisfying particular properties such as connectivity or independence. More formally, can we bound the number of “important independent separators” (or connected as well)? Is it possible to find an equivalent of the so-called Pushing Lemma for this set of separators (Lemma 2.37)?

Combinatorial vs. algebraic proofs. Let us end with probably one of the most complicated questions raised in this manuscript. Most of the proofs provided in this thesis are purely combinatorial proofs. But in Chapter 5, we deal with problems which seem to be easier to consider from an algebraic point of view than from a combinatorial point of view. The original Alon-Saks-Seymour conjecture was an extension of the Graham-Pollack theorem. Recall that Graham and Pollack proved that at least $\ell - 1$ edge-disjoint complete bipartite graphs are needed to cover the edges of K_ℓ . There exist lots of proof for this result, though all of them are purely algebraic proofs. Even worse, no combinatorial proof ensures that K_ℓ cannot be covered by a linear number of edge-disjoint complete bipartite graphs.

Finding a combinatorial proof, even for this weaker result would be very interesting. Indeed, even if algebraic proofs are very nice, a combinatorial proof is more “intuitive”: usually we understand more precisely the structure of the problem with a combinatorial proof rather than with an algebraic proof. A combinatorial proof might help us understand the “true reason” why we cannot do better. Finally, a combinatorial proof that a linear number of edge-disjoint complete bipartite graphs is necessary for covering a clique would probably lead to a breakthrough in the understanding of edge-covering problems in graphs from a combinatorial point of view.



Bibliography

- [1] P. Aboulker and N. Bousquet. Excluding cycles with a fixed number of chords. Technical report, <http://arxiv.org/abs/1304.1718>, 2013. Cited page 10.
- [2] R. Aharoni. Ryser’s conjecture for tripartite 3-graphs. *Combinatorica*, 21:1–4, 2001. Cited page 33.
- [3] V. Alekseev. On the number of maximal independent sets in graphs from hereditary classes. *Combinatorial-algebraic methods in discrete optimization*, NNSU Publishers, Nizhny Novgorod University, pages 5–8, 1991. Cited page 122.
- [4] N. Alon. On the density of sets of vectors. *Discrete Math.*, 46:199–202, 1983. Cited page 72.
- [5] N. Alon. Neighborly families of boxes and bipartite coverings. *Algorithms and Combinatorics*, 14:27–31, 1997. Cited pages 119 and 132.
- [6] N. Alon, G. Brightwell, H. Kierstead, A. Kostochka, and P. Winkler. Dominating sets in k -majority tournaments. *J. Comb. Theory, Ser. B*, 96(3):374–387, 2006. Cited pages 43, 44, 63, 64, and 83.
- [7] N. Alon and I. Haviv. private communication. Cited page 119.
- [8] N. Alon and D. Kleitman. Piercing convex sets and the Hadwiger Debrunner (p,q) -problem. In *Adv. Math.* 96, pages 103–112, 1992. Cited pages 35, 92, and 102.
- [9] N. Alon, J. Pach, R. Pinchasi, R. Radoičić, and M. Sharir. Crossing patterns of semi-algebraic sets. *J. Comb. Theory Ser. A*, 111(2):310–326, August 2005. Cited page 126.
- [10] N. Alon, J. Pach, and J. Solymosi. Ramsey-type theorems with forbidden subgraphs. *Combinatorica*, 21(2):155–170, 2001. Cited page 125.
- [11] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the Association for Computing Machinery*, 42(4):844–856, 1995. Cited page 24.
- [12] K. Amano. Some improved bounds on communication complexity via new decomposition of cliques, 2013. preprint. Cited pages 119, 120, 128, and 130.

- [13] R. Anstee. A survey of forbidden configuration results. *Elec. Journ. of Comb.*, 2013. Cited page 74.
- [14] R. Anstee, L. Rónyai, and A. Sali. Shattering news. *Graphs and Combinatorics*, 18:59–73, 2002. Cited pages 72 and 77.
- [15] B. Aronov, E. Ezra, and M. Sharir. Small-size ϵ -nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, July 2010. Cited page 82.
- [16] P. Assouad. Densité et dimension. *Ann. Inst. Fourier (Grenoble)*, 33:233–282, 1983. Cited page 67.
- [17] D. Auger. Minimal identifying codes in trees and planar graphs with large girth. *Eur. J. Comb.*, 31(5):1372–1384, July 2010. Cited page 78.
- [18] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications, 2nd Edition*. 2008. Cited page 21.
- [19] C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd, 1985. Cited pages 13 and 16.
- [20] S. Bessy and N. Bousquet. Colorful paths for 3-chromatic graphs, 2013. Cited page 10.
- [21] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989. Cited page 64.
- [22] H. Bodlaender, L. Cai, J. Chen, M. Fellows, J.A. Telle, and D. Marx. Open problems in parameterized and exact computation. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation, IWPEC 2006*, 2006. Cited page 45.
- [23] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009. Cited page 27.
- [24] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Cross-composition: A new technique for kernelization lower bounds. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 9 of *Leibniz International Proceedings in Informatics*, pages 165–176, 2011. Cited page 27.
- [25] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001. Cited page 122.
- [26] B. Bollobás and A. Radcliffe. Defect Sauer results. *J. Comb. Theory Ser. A*, 72(2):189–208, November 1995. Cited page 72.
- [27] B. Bollobás and A. Thomason. Proof of a conjecture of Mader, Erdős and Hajnal on topological complete subgraphs. *Eur. J. Comb.*, 19:883–887, 1998. Cited page 91.
- [28] M. Bonamy and N. Bousquet. Recoloring graphs via tree-decompositions. preprint. Cited page 11.
- [29] M. Bonamy and N. Bousquet. Recoloring bounded treewidth graphs. In *LAGOS'13*, 2013. Cited pages 9 and 11.

- [30] M. Bonamy, N. Bousquet, and H. Hocquard. Adjacent vertex-distinguishing edge coloring of graphs. In *to appear in EuroComb'13*, 2013. Cited page 10.
- [31] A. Bondy and U. Murty. *Graph theory with applications*. 1967. Cited page 16.
- [32] M. Bougeret, N. Bousquet, and R. Watrigant. Sparsest problem on chordal graphs, 2013. preprint. Cited page 10.
- [33] N. Bousquet, J. Daligault, and S. Thomassé. Multicut is FPT. In *STOC*, pages 459–468, 2011. Cited pages 9, 24, 45, 46, and 48.
- [34] N. Bousquet, J. Daligault, S. Thomassé, and A. Yeo. A polynomial kernel for multicut in trees. In *STACS*, pages 183–194, 2009. Cited pages 9 and 46.
- [35] N. Bousquet, D. Gonçalves, G. Mertzios, C. Paul, I. Sau, and S. Thomassé. Parameterized domination in circle graphs. In *WG*, pages 308–319, 2012. Cited pages 9 and 10.
- [36] N. Bousquet, D. Gonçalves, G. Mertzios, C. Paul, I. Sau, and S. Thomassé. Parameterized domination in circle graphs. *Theory of Computing Systems*, pages 1–28, 2013. Cited pages 10 and 26.
- [37] N. Bousquet, A. Lagoutte, and S. Thomassé. Clique versus Independent Set, 2013. <http://arxiv.org/abs/1301.2474>, preprint. Cited pages 10, 117, and 122.
- [38] N. Bousquet, A. Lagoutte, and S. Thomassé. The Erdős-Hajnal conjecture for paths and antipaths, 2013. <http://arxiv.org/abs/1303.5205>, preprint. Cited pages 10 and 117.
- [39] N. Bousquet and C. Löding. Equivalence and inclusion problem for strongly unambiguous Büchi automata. In *LATA*, pages 118–129, 2010. Cited page 9.
- [40] N. Bousquet and S. Thomassé. VC-dimension and Erdős-Pósa property. preprint. Cited page 10.
- [41] N. Bousquet and S. Thomassé. Scott's induced subdivision conjecture for maximal triangle-free graphs. *Combinatorics, Probability and Computing*, 21:512–514, 2012. Cited pages 10, 63, 64, 88, and 89.
- [42] S. Brandt and S. Thomassé. Dense triangle-free graphs are four colorable: A solution to the Erdős-Simonovits problem. to appear. Cited page 84.
- [43] L. Brunetta, M. Conforti, and M. Fischetti. A polyhedral approach to an integer multicommodity flow problem. *Discrete Applied Mathematics*, 101(1-3):13 – 36, 2000. Cited page 45.
- [44] K. Cameron, E. M. Eschen, C. T. Hoàng, and R. Sritharan. The complexity of the list partition problem for graphs. *SIAM J. Discrete Math.*, 21(4):900–929, 2007. Cited page 119.
- [45] J. Chalopin, L. Esperet, F. Kardos, Z. Li, P. Ossona de Mendez, and S. Thomassé. A characterization of 2-subdivisions in restricted box graphs. 2013+. Cited page 89.
- [46] L. S. Chandran and F. Grandoni. Refined memorization for vertex cover. *Information Processing Letters*, 93(3):123–131, 2005. Cited page 24.

- [47] I. Charon, O. Hudry, and A. Lobstein. Minimizing the size of an identifying or locating-dominating code in a graph is np-hard. *Theor. Comput. Sci.*, 290(3):2109–2120, 2003. Cited page 78.
- [48] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Comput. Complex.*, 15(2):94–114, 2006. Cited page 46.
- [49] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10:229–249, 1990. Cited page 82.
- [50] C. Chekuri and J. Chuzhoy. Large-Treewidth Graph Decompositions and Applications. *ArXiv e-prints*, April 2013. Cited page 31.
- [51] J. Chen, J. Fan, I. Kanj, Y. Liu, and F. Zhang. Multicut in trees viewed through the eyes of vertex cover. *J. Comput. Syst. Sci.*, 78(5):1637–1650, September 2012. Cited pages 9 and 46.
- [52] J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009. Cited pages 46, 48, 51, 53, and 55.
- [53] J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC ’08, pages 177–186, New York, NY, USA, 2008. ACM. Cited pages 47 and 48.
- [54] V. Chepoi, B. Estellon, and Y. Vaxès. Covering planar graphs with a fixed number of balls. *Discrete & Computational Geometry*, 37(2):237–244, 2007. Cited pages 63, 64, 92, 95, 96, 97, and 102.
- [55] R. Chitnis, M. Hajiaghayi, and D. Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. In *SODA*, pages 1713–1725, 2012. Cited pages 47, 48, 56, 60, and 61.
- [56] M. Chudnovsky. Erdős-hajnal conjecture, a survey. preprint, 2013. Cited page 125.
- [57] M. Chudnovsky, I. Penev, A. Scott, and N. Trotignon. Excluding induced subdivisions of the bull and related graphs. *Journal of Graph Theory*, 71(1):49–68, 2012. Cited page 89.
- [58] M. Chudnovsky and P. Seymour. Excluding paths and antipaths. preprint, 2013. Cited page 125.
- [59] M. Chudnovsky and Y. Zwols. Large cliques or stable sets in graphs with no four-edge path and no five-edge path in the complement. *J. Graph Theory*, 70(4):449–472, July 2012. Cited page 125.
- [60] S. M. Cioabă and M. Tait. Variations on a theme of Graham and Pollak. *Discrete Mathematics*, 313:665–676, 2013. Cited page 118.
- [61] K. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete Comput. Geom.*, 37(1):43–58, January 2007. Cited page 82.
- [62] M.-C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximal integer multiflow: A survey. *European Journal of Operational Research*, 162(1):55–69, 2005. Cited page 45.

- [63] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. Cited page 25.
- [64] M. Cygan, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Clique cover and graph separation: New incompressibility results. In *ICALP (1)*, pages 254–265, 2012. Cited page 46.
- [65] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. Wojtaszczyk. The stubborn problem is stubborn no more (a polynomial algorithm for 3-compatible colouring and the stubborn list partition problem). In *SODA*, pages 1666–1674, 2011. Cited page 119.
- [66] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013. Cited page 46.
- [67] D. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete. *Discrete Mathematics*, 30(3):289 – 293, 1980. Cited page 22.
- [68] G. Dantzig. Application of the simplex method to a transportation problem. *Activity Analysis of Production and Allocation*, pages 359–373, 1949. Cited page 38.
- [69] E. Demaine, M. Hajiaghayi, and D. Marx. Open problems – parameterized complexity and approximation algorithms. Number 09511 in *Dagstuhl Seminar Proceedings*, 2010. Cited page 45.
- [70] R. Diestel. *Graph Theory*. 2010. Cited pages 16, 24, 33, and 36.
- [71] G. Ding, P. Seymour, and P. Winkler. Bounding the vertex cover number of a hypergraph. *Combinatorica*, 14(1):23–34, 1994. Cited pages 63, 64, 74, 80, 82, and 87.
- [72] R. Downey, P. Evans, and M. Fellows. Parameterized learning complexity. In *Proceedings of the sixth annual conference on Computational learning theory*, COLT '93, pages 51–57, New York, NY, USA, 1993. ACM. Cited page 93.
- [73] R. Downey and M. Fellows. Parameterized computational feasibility. In Peter Clote and Jeffrey B. Remmel, editors, *Feasible Mathematics II*, volume 13 of *Progress in Computer Science and Applied Logic*, pages 219–244. Birkhäuser Boston, 1995. Cited page 93.
- [74] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141(1–2):109–131, 1995. Cited page 26.
- [75] R.G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999. Cited pages 22, 26, and 44.
- [76] A. Dress. Towards a theory of holistic clustering. *DIMACS Ser. Discr. Math. Theor. Comput. Sc.*, 37:271–289, 1997. Cited page 72.
- [77] Z. Dvůrák. Constant-factor approximation of the domination number in sparse graphs. *Eur. J. Comb.*, 34(5):833–840, 2013. Cited page 97.
- [78] Z. Dvůrák and D. Král. Classes of graphs with small rank decompositions are x-bounded. *Eur. J. Comb.*, 33(4):679–683, 2012. Cited page 114.
- [79] J. Edmonds. Paths, trees and flowers. *Canad. J. Math*, 17:449–467, 1965. Cited page 32.

- [80] P. Erdős and A. Hajnal. Ramsey-type theorems. *Discrete Applied Mathematics*, 25(1–2):37 – 52, 1989. Cited page 125.
- [81] P. Erdős, A. Hajnal, and J. Pach. Ramsey-type theorem for bipartite graphs. *Geombinatorics*, 10:64:68, 2000. Cited page 126.
- [82] P. Erdős and L. Pósa. On independent circuits contained in a graph. *Canad. J. Math.*, 17:347–352, 1965. Cited page 31.
- [83] P. Erdős and R. Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 35:85–90, 1960. Cited page 144.
- [84] P. Erdős and M. Simonovits. On a valence problem in extremal graph theory. *Discrete Math.*, 5:323–334, 1973. Cited page 84.
- [85] P. Erdős. Graph theory and probability. *Canad. J. Math*, pages 34–38, 1959. Cited page 19.
- [86] P. Erdős. On a problem of graph theory. *Studia Sic Math. Hungar*, 1:215–235, 1966. Cited page 83.
- [87] T. Feder and P. Hell. Full constraint satisfaction problems. *SIAM J. Comput.*, 36(1):230–246, 2006. Cited pages 119 and 135.
- [88] T. Feder, P. Hell, and J. Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003. Cited page 119.
- [89] T. Feder, P. Hell, S. Klein, and R. Motwani. List partitions. *SIAM J. Discrete Math.*, 16(3):449–478, 2003. Cited pages 119, 135, and 138.
- [90] M. Fellows, D. Lokshtanov, N. Misra, M. Mnich, F. A. Rosamond, and S. Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory of Computing Systems*, 45(4):822–848, 2009. Cited page 24.
- [91] S. Fiorini, S. Massar, S. Pokutta, H. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In *STOC*, pages 95–106, 2012. Cited page 118.
- [92] S. Floyd and M. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Mach. Learn.*, 21(3):269–304, December 1995. Cited pages 64 and 74.
- [93] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Text in Theoretical Computer Science. Springer, 2006. Cited pages 22 and 24.
- [94] F. V. Fomin, S. Gaspers, S. Saurabh, and S. Thomassé. A linear vertex kernel for maximum internal spanning tree. In *International Symposium on Algorithms and Computation (ISAAC)*, volume 5878 of *Lecture Notes in Computer Science*, pages 275–282, 2009. Cited page 27.
- [95] F. Foucaud, E. Guerrini, M. Kovse, R. Naserasr, A. Parreau, and P. Valicov. Extremal graphs for the identifying code problem. *Eur. J. Comb.*, 32(4):628–638, 2011. Cited page 78.
- [96] J. Fox and J. Pach. Erdős-Hajnal-type results on intersection patterns of geometric objects. In *Horizons of Combinatorics*, volume 17, pages 79–103. Springer Berlin Heidelberg, 2008. Cited page 126.

- [97] J. Fox and B. Sudakov. Induced ramsey-type theorems. *Advances in Mathematics*, 219(6):1771 – 1800, 2008. Cited page 126.
- [98] J. Fox and B. Sudakov. Density theorems for bipartite graphs and related ramsey-type results. *Combinatorica*, 29(2):153–196, 2009. Cited page 126.
- [99] P. Frankl. On the trace of finite sets. *Journal of Combinatorial Theory, Series A*, 34(1):41 – 45, 1983. Cited pages 71 and 72.
- [100] P. Frankl and J. Pach. On the number of sets in a null t-design,. *Eur. J. Comb.*, 4:21–23, 1983. Cited pages 64 and 72.
- [101] Z. Furedi and J. Pach. Traces of finite sets: Extremal problems and geometric applications. *Extremal problems for finite sets*, 3:251, 1994. Cited page 74.
- [102] Z. Furedi and F. Quinn. Traces of finite sets. *Ars Combinatoria*, 18:195–200, 1984. Cited page 74.
- [103] N. Garg, V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. pages 698–707, 1993. Cited page 46.
- [104] N. Garg, V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997. Cited page 45.
- [105] C. Gavaille, D. Peleg, A. Raspaud, and E. Sopena. Small k-dominating sets in planar graphs with applications. In *Proceedings of WG '01*, pages 201–216, London, UK, UK, 2001. Springer-Verlag. Cited page 96.
- [106] G. Gottlob and S. Tien Lee. A logical approach to multicut problems. *Information Processing Letters*, 103(4):136–141, 2007. Cited page 46.
- [107] R. Graham and H. Pollak. On embedding graphs in squashed cubes. *Graph theory and applications*, 303:99–110, 1972. Cited pages 118 and 128.
- [108] G. Greco. Embeddings and the trace of finite sets. *Inf. Process. Lett.*, pages 199–203, 1998. Cited page 72.
- [109] M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988. Cited page 38.
- [110] S. Guillemot. FPT algorithms for path-transversals and cycle-transversals problems in graphs. In *IWPEC*, pages 129–140, 2008. Cited page 46.
- [111] J. Guo, F. Hüffner, E. Kenar, R. Niedermeier, and J. Uhlmann. Complexity and exact algorithms for multicut. In *SOFSEM*, pages 303–312, 2006. Cited page 46.
- [112] J. Guo and R. Niedermeier. Fixed-parameter tractability and data reduction for multicut in trees. *Networks*, 46(3):124–135, 2005. Cited page 45.
- [113] A. Gyárfás. Problems from the world surrounding perfect graphs. *Zastos. Mat.*, pages 413–441, 1987. Cited pages 89 and 126.

- [114] A. Gyárfás and J. Lehel. Hypergraph families with bounded edge cover or transversal number. *Combinatorica*, 3(3-4):351–358, 1983. Cited page 34.
- [115] H. Hadwiger and H. Debrunner. Über eine variante zum hellyschen satz. *Archiv der Mathematik*, 8:309–313, 1957. Cited page 35.
- [116] D. Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, September 1992. Cited page 64.
- [117] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. In *Symposium on Computational Geometry*, pages 61–71, 1986. Cited pages 63, 80, and 81.
- [118] P. Haxell and A. Scott. On Ryser’s conjecture. *Electr. Journ. of Comb.*, 19, 2012. Cited page 33.
- [119] E. Helly. Über mengen konvexer körper mit gemeinschaftlichen punkten. jahresbericht der deutschen. *Deutsch. Math. Verein*, 32:175–176, 1923. Cited page 35.
- [120] H. Huang and B. Sudakov. A counterexample to the Alon-Saks-Seymour conjecture and related problems. *Combinatorica*, 32:205–219, 2012. Cited pages 118, 128, 129, and 132.
- [121] B. M. P. Jansen and H. L. Bodlaender. Vertex cover kernelization revisited: Upper and lower bounds for a refined parameter. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 9 of *Leibniz International Proceedings in Informatics*, pages 177–188, 2011. Cited page 24.
- [122] J. Kahn. Recent results on some not-so-recent hypergraph matching and covering problems. In *Extremal problems for finite sets*. Bolyai Society mathematical studies, 1994. Cited page 118.
- [123] N. Kakimura, K. Kawarabayashi, and D. Marx. Packing cycles through prescribed vertices. *Journal of Combinatorial Theory, Series B*, 101(5):378 – 381, 2011. Cited page 31.
- [124] G. Kalai and J. Matoušek. Guarding galleries where every point sees a large area. *Israel Journal of Mathematics*, 101:125–139, 1997. Cited pages 64 and 69.
- [125] M. Karpinski and A. Macintyre. Polynomial bounds for VC-dimension of sigmoidal neural networks. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, STOC ’95, pages 200–208, New York, NY, USA, 1995. ACM. Cited pages 64 and 69.
- [126] M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Trans. Inf. Theor.*, 44(2):599–611, September 1998. Cited page 78.
- [127] J. Kim. The Ramsey number $R(3, t)$ has order of magnitude $t^2 / \log(t)$. *Random Structures and Algorithms*, 7:173–207, 1995. Cited page 19.
- [128] J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for ϵ -nets. *Discrete Comput. Geom.*, 7(2):163–173, March 1992. Cited page 81.
- [129] A. Kostochka and X. Zhu. Adapted list coloring of graphs and hypergraphs. *SIAM J. Discrete Math.*, 22(1):398–408, 2008. Cited page 119.
- [130] S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. In *ICALP (1)*, pages 581–593, 2012. Cited pages 47 and 56.

- [131] D. Kühn and D. Osthus. Induced subdivisions in K_s , s -free graphs of large average degree. *Combinatorica*, 24(2):287–304, 2004. Cited page 89.
- [132] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997. Cited page 121.
- [133] D. Kuzmin and M. Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8:2047–2081, 2007. Cited page 74.
- [134] D. König. Gráfok és mátrixok. *Matemat. és Fizikai Lapok*, 38:116–119, 1931. Cited page 32.
- [135] A. Lagoutte and T. Trunck. Clique stable separation in perfect graphs with no balanced skew-partitions, 2013+. preprint. Cited page 128.
- [136] D. Lokshantov and D. Marx. Clustering with local restrictions. *Inf. Comput.*, 222:278–292, 2013. Cited page 56.
- [137] D. Lokshantov and M. Ramanujan. Parameterized tractability of multiway cut with parity constraints. In *Proceedings of the 39th international colloquium conference on Automata, Languages, and Programming - Volume Part I, ICALP'12*, pages 750–761, Berlin, Heidelberg, 2012. Springer-Verlag. Cited page 56.
- [138] L. Lovász. Communication complexity: a survey. In *Paths, flows, and VLSI-layout*, pages 235–265. 1990. Cited page 118.
- [139] L. Lovász. Stable sets and polynomials. *Discrete Mathematics*, 124(1-3):137–153, 1994. Cited pages 118 and 123.
- [140] T. Luczak and S. Thomassé. Coloring dense graphs via VC-dimension. *submitted*. Cited pages 63, 64, 84, 85, and 86.
- [141] B. Lévêque, F. Maffray, and N. Trotignon. On graphs with no induced subdivision of K_4 . *Jour. of Comb. Theory, Ser. B*, 102(4):924 – 947, 2012. Cited page 89.
- [142] W. Mader. Homomorphieeigenschaften und mittlere kantendichte von graphen. *Mathematische Annalen*, 174(4):265–268, 1967. Cited page 91.
- [143] D. Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006. Cited pages 46, 48, 51, 54, and 55.
- [144] D. Marx, B. O’Sullivan, and I. Razgon. Treewidth reduction for constrained separation and bipartization problems. In *STACS*, 2010. Cited page 46.
- [145] D. Marx, B. O’Sullivan, and I. Razgon. Finding small separators in linear time via treewidth reduction. *CoRR*, abs/1110.4765, 2011. Cited page 46.
- [146] D. Marx and I. Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem. In *ESA*, volume 5757 of *Lecture Notes in Computer Science*, pages 647–658. Springer, 2009. Cited pages 46, 149, and 167.
- [147] D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *STOC*, pages 469–478, 2011. Cited pages iii, 45, 46, 47, 48, 55, 57, 60, 141, and 167.

- [148] J. Matoušek. Geometric set systems. In A. Balog, G.O.H. Katona, A. Recski, and D. Sza'asz, editors, *European Congress of Mathematics*, volume 169 of *Progress in Mathematics*, pages 1–27. Birkhäuser Basel, 1998. Cited page 64.
- [149] J. Matoušek. *Geometric Discrepancy (An illustrated guide)*. 1999. Cited page 81.
- [150] J. Matoušek. *Lecture Notes on Discrete Geometry*. 2002. Cited page 80.
- [151] J. Matoušek. Bounded VC-dimension implies a fractional helly theorem. *Discrete & Computational Geometry*, 31(2):251–255, 2004. Cited pages 63, 92, and 102.
- [152] J. Matoušek, R. Seidel, and E. Welzl. How to net a lot with little: small ϵ -nets for disks and halfspaces. In *Proceedings of the sixth annual symposium on Computational geometry*, SCG '90, pages 16–22, New York, NY, USA, 1990. ACM. Cited page 82.
- [153] M. Melcher and K. Reid. Monochromatic sinks in nearly transitive arc-colored tournaments. *Discrete Mathematics*, 310(20):2697–2704, 2010. Cited page 84.
- [154] S. Moran. Shattering-Extremal Systems. Technical report, November 2012. Cited pages 64 and 73.
- [155] J. Mycielski. Sur le coloriage des graphes. *Colloq. Math.*, 3:161–162, 1955. Cited page 19.
- [156] J. Nederlof. Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner tree and related problems. In *Automata, Languages and Programming (ICALP)*, volume 5555 of *Lecture Notes in Computer Science*, pages 713–725, 2009. Cited page 27.
- [157] R. Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, March 2006. Cited page 22.
- [158] R. Niedermeier and P. Rossmanith. On efficient fixed-parameter algorithms for weighted vertex cover. *Journal of Algorithms*, 47(2):63–77, 2003. Cited page 24.
- [159] S. Oum and P. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4):514–528, 2006. Cited page 98.
- [160] J. Pach and P. Agarwal. *Combinatorial Geometry*. Wiley-Intersciences Series in Discrete Mathematics and Optimization, 1995. Cited pages 64, 69, and 81.
- [161] J. Pach and G. Tardos. Tight lower bounds for the size of ϵ -nets. In *Proceedings of the 27th annual ACM symposium on Computational geometry*, SoCG '11, pages 458–463, New York, NY, USA, 2011. ACM. Cited page 82.
- [162] J. Pach and G. Woeginger. Some new bounds for epsilon-nets. In *Proceedings of the sixth annual symposium on Computational geometry*, SCG '90, pages 10–15, New York, NY, USA, 1990. ACM. Cited page 81.
- [163] D. Pálvölgyi and A. Gyárfás. Domination in transitive colorings of tournaments. *ArXiv e-prints*, February 2013. Cited pages 83 and 84.
- [164] C. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the VC-dimension. *J. Comput. Syst. Sci.*, 53(2), 1996. Cited page 94.

- [165] A. Pawlik, J. Kozik, T. Krawczyk, M. Lasoń, P. Micek, W. T. Trotter, and B. Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *ArXiv e-prints*, September 2012. Cited pages 19 and 89.
- [166] M. Pontecorvi and P. Wollan. Disjoint cycles intersecting a set of vertices. *Journal of Combinatorial Theory, Series B*, 102(5):1134 – 1141, 2012. Cited page 31.
- [167] E. Pyrga and S. Ray. New existence proofs for ϵ -nets. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, SCG '08, pages 199–207, New York, NY, USA, 2008. ACM. Cited page 82.
- [168] L. Linial, R. Aharoni and R. Meshulam. On the trace of incompatible vectors. *Congressus Numerantium*, 113:15–18, 1996. Cited page 72.
- [169] D. Rautenbach and B. Reed. The erdős-pósa property for odd cycles in highly connected graphs. *Combinatorica*, 21:267–278, 2001. Cited page 32.
- [170] A. Razborov. The gap between the chromatic number of a graph and the rank of its adjacency matrix is superlinear. *Discrete Mathematics*, 108(1-3):393–396, 1992. Cited page 128.
- [171] I. Razgon and B. O’Sullivan. Almost 2-SAT is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75:435–450, 2009. Cited page 167.
- [172] B. Reed, N. Robertson, P. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996. Cited page 32.
- [173] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32:299–301, 2004. Cited page 24.
- [174] V. Rödl. On universality of graphs with uniformly distributed edges. *Discrete Mathematics*, 59(1–2):125 – 134, 1986. Cited page 125.
- [175] R. Rizzi. Minimum weakly fundamental cycle bases are hard to find. *Algorithmica*, 53(3):402–424, February 2009. Cited page 31.
- [176] N. Robertson and P. Seymour. Graph minors V. Excluding a planar graph,. *J. Combin. Theory Ser. B*, 41, 1986. Cited page 31.
- [177] B. Sands, N. Sauer, and R. Woodrow. On monochromatic paths in edge-coloured digraphs. *Jour. of Comb. Theory, Ser. B*, 33:271–275, 1982. Cited page 84.
- [178] N. Sauer. On the density of families of sets. *J. Comb. Theory, Ser. A*, 13(1):145–147, 1972. Cited pages 63 and 72.
- [179] D. Schell. Matrix partitions of digraphs, 2009. Simon Fraser University. Cited page 119.
- [180] A. Schrijver. *Combinatorial Optimization*. 2004. Cited pages 37 and 43.
- [181] A. Scott. Induced trees in graphs of large chromatic number. *J. Graph Theory*, 24:297–311, 1997. Cited page 89.
- [182] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pac. J. Math.*, 41:247–261, 1972. Cited page 72.

- [183] C. Thomassen. On the presence of disjoint subgraphs of a specified type. *Journal of Graph Theory*, 12(1):101–111, 1988. Cited page 31.
- [184] C. Thomassen. On the chromatic number of triangle-free graphs of large minimum degree. *Combinatorica*, 22:591–596, 2002. Cited page 84.
- [185] C. Thomassen. On the chromatic number of pentagon-free graphs of large minimum degree. *Combinatorica*, 27(2):241–243, 2007. Cited page 86.
- [186] N. Trotignon and K. Vušković. A structure theorem for graphs with no cycle with a unique chord and its consequences. *Journal of Graph Theory*, 63(1):31–67, 2010. Cited page 89.
- [187] Z. Tuza. Ryser’s conjecture on transversals of r -partite hypergraphs. *Ars Combinatoria*, 16:201–209, 1983. Cited page 33.
- [188] H. Tverberg. On the decomposition of kn into complete bipartite subgraphs. *J. Graph Theory*, 6:493–494, 1982. Cited page 128.
- [189] V. Vapnik. *Estimation of dependences based on empirical data*. 1982. Cited page 64.
- [190] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971. Cited pages 64 and 72.
- [191] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. Syst. Sci.*, 43(3):441–466, 1991. Cited pages 117, 118, and 120.
- [192] A. Yeo. private communication. Cited page 130.
- [193] J. Zaks. Bounds on neighborly families convex polytopes. *Geometriae Dedicata*, 8:279–296, 1979. Cited page 132.
- [194] A. Zykov. On some properties of linear complexes (in russian). *Math. Sbornik.*, 24:163–188, 1949. Cited page 19.



Index

- (p, q) -property, 35
- 2-shattered, 86
- 2VC-dimension, 87
- B -hypergraph, 21
- $B(x, k)$, 19
- B_ℓ -hypergraph, 21
- F -free, 16
- H_t -free, 74
- T -compatible set, 56
- $W[1]$ -complete, 26
- Δ -system, 144
- Δ , 48
- α , 17
- bp**(G), 128
- χ , 19
- δ , 48
- ϵ -net, 80
- $\mathcal{C}_{k,n}$, 16
- \mathcal{F} -hypergraph, 56
- $\mathcal{U}_{k,n}$, 15
- $v(H)$, 30
- v^* , 42
- ω , 17
- \bar{X} , 13
- τ , 29
- τ^* , 42
- k -edge coloring, 17
- k -majority tournament, 70
- k -partite, 32
- k -uniform, 15
- st -path, 33
- st -separator, 33
- χ -bounded, 89
- active, 147
- acyclic, 20
- adjacent, 16
- antichain, 13
- arc, 21
- auto-dual, 15
- backbone, 148
- bipartite, 20
- border, 48
- chain, 13
- cherry, 146
- chromatic number, 19
- circuit, 21
- CL-IS, 118
- clique, 17
- collapse, 76
- collapse index, 76
- coloring, 19
- comparable, 13
- complement, 15, 16
- complete hypergraph, 16

- complete Venn diagram, 66
- connected, 16, 20
- connected component, 20
- connectivity, 49
- constraint, 37
- constraint matrix, 38
- covering, 36
- covers V , 13
- critical edge, 150
- CS-separator, 121
- cube, 84
- cut, 117
- cycle, 19

- decision problem, 22
- degree, 16
- deletion allowance, 143
- directed graph, 21
- dominating set, 21
- dual, 14
- dual hypergraph, 14
- dual paired VC-dimension, 85
- dual VC-dimension, 66

- ellipsoid method, 38
- endpoints, 16
- Erdős-Pósa property, 31
- Erdős Hajnal conjecture, 125
- exact shadow, 57

- Farkas' Lemma, 43
- feedback vertex set, 21
- fooling set, 121
- forest, 20
- FPT, 22
- fractional relaxation, 39
- free connectivity, 153
- free edges, 149

- gap function, 31
- girth, 19
- graph, 14

- half-requests, 152
- half-space, 37

- Helly property, 33
- hereditary, 16
- hitting set, 29
- hyperedge, 13
- hypergraph, 14
- hyperplane, 37

- identifying code, 78
- ILP, 38
- important separator, 49
- incidence bipartite graph, 14
- indivisible separator, 49
- induced copy, 16
- integrality gap, 39
- interior, 19
- irrelevant, 146

- kernel, 26

- leaves, 20
- left clean, 157
- lemon, 159
- length, 19
- LP, 37
- LP-dual, 38

- matching, 32
- minimum path, 19
- multicut, 45

- negative, 22
- neighbor, 16
- neighborhood, 16
- nodes, 20

- objective function, 38
- opposite hypergraph, 15
- order, 13
- ordered shattered set, 75
- oriented, 21

- packing, 30
- paired hypergraph, 85
- parameter, 22
- path, 19
- polyhedron, 37

polynomial kernel, 27
polytope, 37
positive, 22
primal, 38
Pushing Lemma, 54

requests, 45
root, 48

Sandwich Theorem, 71
Sauer's Lemma, 72
separator, 48
shadow, 56
shadow removal, 57
shattered, 64
simple, 14
slice, 150
split, 20
stable set, 17, 36
stable set hypergraph, 20
strongly shattered, 66
subgraph of G , 16
subhypergraph, 14

tag, 155
terminals, 45
Theorem of Ding-Seymour-Winkler, 87
Theorem of Haussler Welzl, 80
tournament, 21
trace, 14
transitive tournament, 21
transversal, 29
transversality, 29
tree, 20
triangle, 17

underlying graph, 21
uniform measure, 80

VC-dimension, 64
vector, 37
vertex cover, 32
vertex set, 13
vertices, 13

walk, 19



List of Figures

0.1	Non equality between minimum hitting set and maximum packing.	4
1.1	A hypergraph	14
1.2	The incidence bipartite graph	14
1.3	Dual hypergraph of Figure 1.1	15
1.4	Complement hypergraph of Figure 1.1	15
1.5	Representation of $\mathcal{U}_{3,4}$	16
1.6	Several graphs of size 4	17
1.7	A net	20
1.8	A graph which admits a vertex cover of size 2.	25
2.1	Hitting set	30
2.2	Packing	30
2.3	Escher's wall of height 4	31
2.4	Helly property for rectangles	34
2.5	Hypergraphs of rectangles does not satisfy the Helly property	34
2.6	(3,2)-property	35
2.7	A 2-dimensional polytope defined by 4 constraints.	37
2.8	Gap between LP and ILP	39
2.9	A hypergraph	40
2.10	Constraint matrix of the Transversal LP	40
2.11	Gap between packing and fractional packing	43
2.12	An exponential number of separators	48
2.13	Illustration of Lemma 2.28.	50
2.14	The importance of indivisible important separators	53
2.15	A graph with almost $\mathcal{O}^*(4^k)$ indivisible important YX -separators of size k . The sets T' and I are illustrations of the proof of Lemma 2.35.	53
2.16	Shadow of a \mathcal{F} -HITTING SET	56

2.17	Exact shadows	57
2.18	Importance of exact shadows	59
2.19	Illustration of the proof of Lemma 2.42	59
3.1	A shattered set	64
3.2	Complete Venn diagram	67
3.3	Shattered sets by d -dimension rectangles	69
3.4	A 2-majority tournament	70
3.5	A hypergraph such that $sh(H) \neq E $	73
3.6	The opposite hypergraph of Figure 3.5	73
3.7	Ordered shattered set	75
3.8	An identifying code	78
3.9	Epsilon-net	79
3.10	Tournament union of 3 partial orders with large VC-dimension	83
3.11	A cube	84
3.12	2-complete Venn diagram	86
3.13	A 1-subdivided K_4	88
3.14	Construction for the proof of Theorem 3.33	90
3.15	Large 2VC-dimension for neighborhood of interval graphs	91
4.1	The graph $G_{n,\ell}$ of Theorem 4.12 with $n = 4$ and $\ell = 2$	100
4.2	4 types of crosses.	104
4.3	Minimum paths with root sections (dashed parts), critical vertices and precritical vertices.	107
4.4	Examples of escapes. The right one is an edge of the $a'b$ -path.	109
4.5	Jump paths	112
4.6	Illustration of Claim 4.37	114
5.1	Matrix M for the stubborn problem.	119
5.2	A clique K and a stable S in G	124
5.3	Graph B built from K and S . Edges of G are replaced by forward arcs, and non-edges are replaced by backward arcs.	124
5.4	Hypergraph H where hyperedges are built from the non-neighborhood of vertices from S	124
5.5	A graph satisfying $\mathbf{bp}_{\text{or}} \neq \mathbf{bp}$	129
5.6	An instance of 3-CCP	134
5.7	Solution of the 2-CCP instance with a compatible 2-list assignment	134
5.8	Another solution to the instance with a compatible 2-list assignment.	134
5.9	Illustrations of definition of 2-list assignment	134
5.10	Diagram representing the stubborn problem	134
5.11	Illustration of the proof of Lemma 5.25	137
5.12	Proof the stubborn problem implies 3-ALCP (1)	138
5.13	Proof the stubborn problem implies 3-ALCP (2)	138
5.14	Proof the stubborn problem implies 3-ALCP (3)	138
5.15	Proof the stubborn problem implies 3-ALCP (global)	138
5.16	Illustration of the proof of Lemma 5.27	139

5.17	Illustration of the proof of Lemma 5.28	140
6.1	Links between COMPONENT MULTICUT and BACKBONE MULTICUT	153
6.2	A component in a BACKBONE MULTICUT	154
6.3	Illustration of backbones and slices	155
6.4	Full backbone vertices	156
6.5	Lemonizing components	158
6.6	The left subgraph L_i of v_i	159
6.7	Transformation into lemons for VERTEX MULTICUT	166
6.8	A double path on seven vertices.	168
6.9	Illustration for Conjecture 8	168

Abstract

In this manuscript we study hitting sets both from a combinatorial and from an algorithmic point of view. A *hitting set* is a subset of vertices of a hypergraph which intersects all the hyperedges. A *packing* is a subset of pairwise disjoint hyperedges. In the general case, there is no function linking the minimum size of a hitting set and a maximum size of a packing.

The first part of this thesis is devoted to present upper bounds on the size of hitting sets, in particular these upper bounds are expressed in the size of the maximum packing. Most of them are satisfied when the dimension of Vapnik-Chervonenkis of the hypergraph is bounded. The originality of this thesis consists in using these hypergraph tools in order to obtain several results on graph problems. First we prove that a conjecture of Scott holds for maximal triangle-free graphs. Then we generalize a result of Chepoi, Estellon and Vaxès on dominating sets at large distance. We finally study a conjecture of Yannakakis and prove that it holds for several graph subclasses using VC-dimension.

The second part of this thesis explores algorithmic aspects of hitting sets. More precisely we focus on parameterized complexity of graph separation problems where we are looking for hitting sets of a set of paths. Combining connectivity tools, important separator technique and Dilworth's theorem, we design an FPT algorithm for the MULTICUT problem parameterized by the size of the solution.

Keywords: *Hitting set, packing, Erdős-Pósa property, VC-dimension, parameterized complexity, FPT algorithm, important separators, graph separation problems*

Résumé

Dans cette thèse, nous étudions des problèmes de transversaux d'un point de vue tant algorithmique que combinatoire. Étant donné un hypergraphe, un *transversal* est un ensemble de sommets qui touche toutes les hyperarêtes. Un *packing* est un ensemble d'hyperarêtes deux à deux disjointes. Alors que la taille minimale d'un transversal est au moins égale à la taille maximale d'un packing on ne peut pas dans le cas général borner la taille minimale d'un transversal par une fonction du packing maximal.

Dans un premier temps, un état de l'art rappelle les différentes conditions qui assurent l'existence de bornes supérieures sur la taille des transversaux, en particulier en fonction de la taille d'un packing. La plupart d'entre elles sont valables lorsque la VC-dimension de Vapnik-Chervonenkis de l'hypergraphe, est bornée. L'originalité de la thèse consiste à utiliser ces outils d'hypergraphes pour obtenir des résultats sur des problèmes de graphes. Nous prouvons notamment une conjecture de coloration de Scott dans le cas des graphes sans-triangle maximaux; ensuite, nous généralisons un résultat de Chepoi, Estellon et Vaxès traitant de domination à grande distance; enfin nous nous attaquons à une conjecture de Yannakakis sur la séparation des cliques et des stables d'un graphe.

Dans un second temps, nous étudions les transversaux d'un point de vue algorithmique. On se concentre plus particulièrement sur les problèmes de séparation de graphe où on cherche des transversaux à un ensemble de chemin. En combinant des outils de connexité, les séparateurs importants et le théorème de Dilworth, nous obtenons un algorithme FPT pour le problème MULTICUT paramétré par la taille de la solution.

Mots clefs : *Transversal, packing, propriété d'Erdős-Pósa, VC-dimension, complexité paramétrée, algorithme FPT, séparateurs importants, problèmes de séparation de graphe*
