HELLENIC REPUBLIC
National and Kapodistrian
University of Athens

Archontia C. Giannopoulou

Partial Orderings and Algorithms on Graphs

PhD Thesis

Advisor: Dimitrios M. Thilikos

Department of Mathematics
October 2012

Abstract

In the Graph Minors project, N. Robertson and P. Seymour, proved a series of structural and algorithmic results. Some of them, such as the Strong and the Weak Structure Theorem, the Excluded Grid Theorem and the algorithm for MINOR CONTAINMENT, constitute a rich source of many more structural as well as algorithmic results.

Furthermore, the development of Graph Minor Theory coincided with and influenced the development of a new branch of Complexity, namely the Parameterized Complexity Theory, introduced by R. Downey and M. Fellows.

In this doctoral thesis we deal with a series of issues in Graph Minor Theory and Parameterized Complexity as well as the dependence of these two areas.

Περίληψη

Στις εργασίες Ελασσόνων Γραφημάτων, οι N. Robertson και P. Seymour απέδειξαν μια σειρά από δομικά και αλγοριθμικά αποτελέσματα. Κάποια από αυτά, όπως το Ισχυρό και το Ασθενές Δομικό Θεώρημα, το Θεώρημα Απαγορευμένης Σχάρας καθώς και ο αλγόριθμος για την ΠΕΡΙΕΚΤΙΚΟΤΗΤΑ ΕΛΑΣΣΟΝΟΣ, αποτελούν πλούσια πηγή πολλών ακόμα δομικών αλλά και αλγοριθμικών αποτελεσμάτων.

Επιπλέον, η ανάπτυξη της Θεωρίας Ελασσόνων Γραφημάτων συνέπεσε με και επηρέασε την ανάπτυξη ενός νέου κλάδου της πολυπλοκότητας, την Θεωρία Παραμετρικής Πολυπλοκότητας, που προτάθηκε από τους R. Downey και M. Fellows.

Σε αυτή την διδακτορική διατριβή ασχολούμαστε με μια σειρά από ζητήματα της Θεωρίας Ελασσόνων Γραφημάτων και της Θεωρίας Παραμετρικής Πολυπλοκότητας καθώς και με την αλληλεξάρτηση των δύο αυτών περιοχών.

Preface

When I began writing this preface I realized that I would really need some space in order to describe my experience at the Department of Mathematics of the National and Kapodistrian University of Athens (UoA), which has hosted me academically from September 2003, when I first started my undergraduate studies, until today. Many people contributed in the successful completion of my studies as well as in my character's growth these nine years.

At this point I wish to thank them, starting with my Advisor, Prof. Dimitrios M. Thilikos. His contribution was crucial during the course of my studies even since my undergraduate years. His professionalism, affability, and perseverance made our collaboration particularly pleasant and, on his behalf, impeccable. He constitutes the exemplar of a Teacher. Furthermore, our mathematical (and not only) conversations as well as his numerous advices, which I deeply appreciate, encouraged me but also averted me from many pitfalls, at the point where the title Academic

"Parent" is far more accurate than the title of the Advisor.

I also wish to warmly thank the other two members of my Three-Member Advising Committee, Prof. Lefteris M. Kirousis and Prof. Stavros G. Kolliopoulos. I would like to say an extra *thank you* to Prof. Kirousis for the interest he showed in the Department's Forum.

Continuing, I wish to thank Prof. Christos A. Athanasiadis, Prof. Constantinos Dimitracopoulos, Prof. Evagelos Raptis and Prof. Daniel Lokshtanov for doing me the honor of participating to the Seven-Member Examining Committee of my PhD thesis. Special thanks I want to give, alphabetically, to Prof. Dimitracopoulos, who was the Head of the Master's Program in Logic, μΠλ∀[1] during my studies there, Prof. Lokshtanov, for personally covering all his expenses for his travel and stay in Greece the week that the PhD thesis' defence took place, and Prof. Raptis, for his impeccable cooperation at the Computer Laboratory and the Forum of the Department.

I also thank the Department of Mathematics of UoA for the financial support of my PhD studies by offering me a grant under the research project 70/4/10311 of the Special Account for Research Grants of UoA.

Continuing, I wish to thank, chronologically, the Professors who hosted me and supported me financially while visiting them. In particular, I would like to thank:

- Prof. Isolde Adler, for hosting me at Goethe Universität, Frankfurt am Main and her home, her financial (and not only) support as well as her endless willingness to answer any of my questions and generously discuss any research subject during my stay in Frankfurt in October of 2010.
- Prof. Saket Saurabh, for his full (including financial) support and his

[1] Inter - University Program of Graduate Studies in Logic and Theory of Algorithms and Computation

encouragement to share any thought or doubt, during my visit in the Institute of Mathematical Sciences, Chennai in India at spring of 2011. His ability to make everything seem easy, as well as the tenacity according to which he does research have been and are a great source of inspiration.

• Prof. Fedor V. Fomin, for openhandedly covering all the expenses of my trip to Bergen at spring of 2011, enabling me to visit the Algorithms Group of Universitetet i Bergen and attend Treewidth Workshop 2011. I also wish to warmly thank him for his trust and support. I look forward to taking my next research step as part of his research group.

• Prof. Daniel Král' and Prof. Zdeněk Dvořák for financially supporting my visit at Univerzita Karlova v Praze in Prague during the fall and winter of 2011. I also owe a special *děkuji* to Prof. Jaroslav Nešetřil for encouraging me since my first research steps.

At this point, I would like to thank μΠλ∀ and all of my Professors there as they offered me the knowledge and the motivation that made all of the above possible. I would like to particularly thank Prof. Yiannis N. Moschovakis, who is the model of scientist, not just for me but also for the majority (if not all) of his students.

Taking a small detour, I would like to mention that there were some times where my efforts to tackle the bureaucratic issues of the Department arrived to a dead end. I wish to kindly thank the Head of the Department of Mathematics, Prof. Ioannis Emmanouil, as well as the Head of the Committee of the Program of Post-graduate Studies, Prof. Antonios Melas, for their promptitude and effectiveness towards resolving some of them.

I should not neglect to thank Dimitris Zoros and Athanassios Koutsonas for making our working environment pleasant during sharing of the room 116 and the rest of my fellow students that made our common stay in the Department of Mathematics interesting and pleasant.

I also thank all of my collaborators that I have not already explicitly thanked. These are Marcin Kamiński, to whom I am also indebted for his support and kindness, and finally Sudeshna Kolay and Iosif Salem.

Continuing, I thank the members of the Computer Laboratory and the Forum of the Department for the pleasant hours I spent there, as well as LabRadio and its chat for accompanying me in many hours of studying and resting.

I also wish to thank all of my friends who visited me during my stays abroad and who invited me to visit them during their stays abroad. Their presence is always welcomed.

Finally, I would like to thank my parents for the unconditional support of all the choices that I have ever made.

«Να φεύγω τώρα είναι η ώρα κι ίσως σας ξαναδώ ...»[2]

Archontia C. Giannopoulou
October 2012
Athens

[2]These are lyrics from a song written by Pavlos Sidiropoulos. Loosely translated, they say that: "I am going away now, the time has come, but perhaps I will see you again".

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 In General ...

Let us imagine a room occupied by n persons where some of them, say m, pairwise mutually know each other while the rest are strangers. This everyday situation, as well as numerous others, is modeled by the mathematical notion of a graph.

More specifically, a graph G is an ordered pair of two sets (V, E), where V is called *vertex set* of the graph and E is called *edge set* and consists of 2-subsets of V.

While the first result in Graph Theory appeared in 1736, with the negative solution of the Bridges of Königsberg problem by L. Eüler, the development of this theory did not start up until later and the notion of a graph was not formally defined until 1878 [218]. Nevertheless, its development was rapid during the last century, and especially during the last decades.

The participation of Graph Theory in Mathematics is now particularly strong both in Combinatorics and Discrete Mathematics as well as in the Theory of Algorithms.

In this doctoral thesis, we study a series of issues in Graph Theory related to Graph Minors Theory and to Parameterized Complexity Theory, and try to contribute a piece of sand in the enterprises of these Theories. (From now on, we assume that the reader has elementary knowledge of the notions of Graph Theory and of the Theory of Algorithms.)

1.2 . . . and More Specifically

The Graph Minors Theory [185–194, 198–202, 202–209] was developed by N. Robertson and P. Seymour towards the resolution of a conjecture, known as Wagner's Conjecture. The motivation of this conjecture was a classical theorem, Kuratowski's Theorem [142], which provided a complete characterization of planar graphs in terms of forbidden topological minors, and its subsequent restatement, by Wagner [221], in terms of forbidden minors. Wagner's Conjecture, now known as the Robertson-Seymour Theorem [194], claimed that in every infinite family of graphs there are two of them such that one contains the other as a minor.

The results of this theory cleared the picture in regard to the classes of graphs that forbid some fixed graph as a minor, and constitute both the source of inspiration of and a powerful instrument for the proofs of many more results in Structural [83, 123–125, 196], as well as in Algorithmic [5, 51, 86, 90, 212], Graph Theory. One of the most important results, in both of these fields, was the corollary that every graph class that is closed under taking of minors, that is, if a graph belongs to this graph class, then all of its minors also belong to the graph class, can be characterized by a finite set of forbidden graphs. For example, it is known that, for the graph class of planar graphs, these forbidden graphs are K_5 and $K_{3,3}$. As we mentioned a while ago, the importance of this result is duple. In the Theory of Algorithms, its importance lies to the fact that in order to decide whether a graph belongs to a graph class that is closed under taking of minors, it is enough to check whether this graph contains one of the forbidden graphs as a minor.

The proof of this corollary further facilitated the classification of many NP-complete problems. The way was the following. In the framework of the development of Graph Minors Theory, there was given a way to construct, for any fixed graph H, a cubic time algorithm such that for every graph G decides whether H is a minor of G (where the hidden constants depend only on the forbidden graph H) [193]. This algorithmic result immediately implies the existence of a cubic time algorithm that decides whether a graph G belongs to a graph class that is closed under taking of minors (where, as above, the hidden constants depend only on the graph class).

However, this is just one, trivial, corollary/application of the Graph Minors Theory to the Theory of Algorithms. But it has a major algorithmic drawback. While only the existence of such an algorithm is known and, since the Graph Minors Theory is non-constructive [95], no way towards its construction is provided, the set of forbidden graphs is, in general, not known. Thus, in order to construct such an algorithm we first need to identify the forbidden graphs of a class that is closed under taking of minors.

Here, we study the parameter of tree-depth of a graph. Is it known that, the class of graphs that have tree-depth bounded by some constant k, $\mathcal{G}_k$, is closed under taking of minors. We show that, the size of the forbidden minors for the graph class $\mathcal{G}_k$ is bounded by a function on k of order $2^{2^{k-1}}$. Then, we give a precise structural characterization of all the acyclic forbidden graphs of this graph class and prove that they are exactly $\frac{1}{2}2^{2^{k-1}-k}(1+2^{2^{k-1}-k})$. Moreover, we identify exactly all the forbidden minors for $k \in \{1,2,3\}$. (For the case where $k = 4$ it is very easy to see that the number of the graphs is already discouraging.) Finally, we prove a theorem which suggests a procedure such that, starting from a graph G whose tree-depth we are trying to compute, we may obtain a graph of smaller size that has the same tree-depth as G. As we will see later on such rules are particuarly important to the Parameterized Complexity Theory.

Before we continue with the analysis of the results and the applications of the Graph Minors Theory with respect to the minor relation, we would like to take a short detour and discuss the immersion relation as it is also going to be a subject of our study.

In the last paper of the Graph Minors series [211], it was proved that, similarly to the minor relation, an infinite family of graphs contains two graphs that are comparable with respect to the immersion relation.

This result was previously known as the Nash-Williams' Conjecture and its implications to the classes of graphs that are closed under taking immersions are an analogue to the aforementioned ones for the graph classes that are closed under taking of minors. The only difference is that, the algorithmic application requires the recent algorithmic results of M. Grohe, K. Kawarabayashi, D. Marx and P. Wollan in [109].

Subsequently, having been disappointed by the possible size of a set of forbidden graphs, we try to devise more efficient mechanisms for their identification: Algorithms. Our main goal is to answer the following question. Which information of a graph class that is closed under taking of immersions are necessary in order to construct an algorithm that computes its forbidden immersions?

Based on the framework that was devised by I. Adler, M. Grohe, and S. Kreutzer in [4] for the case of minors, we prove that the necessary conditions are an upper bound on the tree-width of the forbidden immersions and a description of the graph class in Monadic Second Order Logic. Moreover, by computing such an upper bound on the tree-width of the obstructions of a graph class that is the union of two graph classes, that are closed under taking of immersions, whose forbidden graphs are known, we (also) propagate the notion of computability in graph classes that are finite unions of classes of graphs that are closed under taking of immersions and whose forbidden graphs are known.

Let us continue, however, with the celebrated results of the Graph Minors Theory with respect to the minor ordering. Choosing the most influential one would be a very challenging task. We would, however, dare to claim that if one of them could assert the first place, then this is the (Strong) Structure Theorem that appeared in GM XVI [207] and reveals the structure of a graph that excludes some other graph as a minor. Its statement requires some quite complicated notions and cannot be given in the context of a more general introduction. Roughly, we could say that, if a graph G does not contain a graph H as a minor, then G has a tree-like, in the topological sense, structure and can be decomposed into smaller graphs which can almost be embedded in a surface in which H cannot be embedded.

This result constitutes both a point of reference for the proof of new structural characterizations of graphs and a useful instrument for designing many algorithms.

Inspired by the Strong Structure Theorem of the Graph Minors Theory, we prove a structural characterization of the graphs that forbid a fixed graph H as an immersion and can be embedded in a surface of Eüler genus γ. In particular, we prove that a graph G that forbids some graph H as an immersion and is embedded in a surface of Eüler genus γ has either "small" tree-width (bounded by a function of H and γ) or "small" edge-connectivity (bounded by the maximum degree of H).

Evenmore, inspired by Kuratowski's Theorem and Wagner's Theorem we give a precise structural characterization of the graphs that do not contain K_5 and $K_{3,3}$ as immersions. In particular, we show that if a graph G contains neither K_5 nor $K_{3,3}$ as an immersion then in can be constructed by repetitive unions of simpler graphs, starting either from graphs of small decomposability (of branch-width at most 10) or from planar graphs of maximum degree at most 3. To show this, we prove an intermediate result on the confluence of a family of edge-disjoint paths.

Continuing the reference in Graph Minors Theory, we would like to mention another structural result, which we could probably say that it is of greater importance to the Theory of Algorithms than the Strong Structure Theorem. This is the Weak Structure Theorem. The Weak Structure Theorem was proved for the first time in GM XIII [193] and was the key-theorem for the construction of the algorithm that decides the containment of a graph in another graph as a minor.

According to the Weak Structure Theorem, a graph G that does not contain a graph H as a minor and has "sufficiently large" tree-width contains, after the removal of a "small" number of vertices, a flat wall as a subgraph. The notion of flatness cannot be easily defined at this point but we can temporarily think that the wall has been arranged in a planar manner inside the graph.

A huge amount of algorithmic applications is derived from this theorem [4, 42, 88, 90, 106, 116–118, 120–122, 127, 130, 131, 133, 134, 138]. The reason is that a technique, namely the irrelevant vertex technique, that was introduced by N. Robertson and P. Seymour indicates that we can remove a vertex from the graph, and more specifically the "middle" vertex of the wall, and end up to an equivalent instance. For more applications using the broader concept, see [44, 85, 115, 154].

Then, the idea behind the algorithm is the following. If we have a graph G that does not contain a graph H as a minor then we check whether it has "sufficiently large" tree-width. In this case, the graph contains a "big enough" wall from which we may remove a specific vertex and obtain an equivalent instance of smaller size. Otherwise, the graph has small tree-width and we can solve the problem using dynamic programming.

It is easy to see that in order to apply the Weak Structure Theorem on algorithmic problems we wish for the relation between the height of the wall and the lower bound on the tree-width of the graph in order to ensure the existence of the wall to be optimal.

For this reason, we prove an optimized version of the Weak Structure Theorem, in which both the number of vertices that we need to remove in order to find the flat wall and the relation between the tree-width of the graph and the height of the wall are linear, and therefore optimal.

Furthermore, using this theorem and two already known results, we discuss how the duality of some tilings of the plane can be "extended" to the realm of the graphs that forbid some fixed apex graph as a minor.

Returning to Graph Minors Theory, we turn our attention to some of the theories that were encouraged and inspired by its results. The first one of them is the Parameterized Complexity Theory [66, 82, 172]. In Parameterized Complexity Theory, the input of every algorithmic problem is a pair whose first component is the input graph while the second component is a function, which is called parameterization of the problem. The aim of the Parameterized Complexity Theory is to examine and classify the time dependence of the solution of the algorithmic problem with respect to each one of their parameterizations. According to M. Langston, one of the first results that led towards this direction, was the result that every graph class that is closed under taking of minors can be decided in cubic time [22]. This made clear that some NP-complete problems can be solved in cubic time when the parameter is fixed, while others are not expected to behave in such a "good manner".

Another theory, based on structural theorems of the Graph Minors, is the Bidimensionality Theory [50, 83]. Bidimensionality Theory is based on theorems according to which a graph G, that forbids another graph H and has "sufficiently large" tree-width, also contains a graph on which the parameter is "bidimensional". Many times, this acts as a certificate according to which the dependence of the running time of a parameterized problem is "good".

We analyze these notions and show how we can apply the optimized version of the Weak Structure Theorem in order to design fixed parameter tractable algorithms. We also use it to complement the Bidimensionality "picture" beyond the relations of minors and contractions, on which it has already been established, to graph parameters that are closed under (distance) topological minors.

Let us now notice that, all algorithmic solutions that we have mentioned up to now concern either specific kinds of problems (for example, the decision of graph classes that are closed under taking of minors) or problems whose input is restricted (for example, the input is some graph

G that does not contain some fixed graph as a minor). Many times this restrictions, indeed, allow us to design algorithms with better running times than in the case of general graphs. However, we would also like to obtain solutions on problems whose input is any graph.

In this thesis, we deal with the problem of max cut on hypergraphs, that is, with the problem of set splitting. In this problem, we are given a hypergraph and our goal is to color its vertices using two colors in such a way that the biggest possible number of its hyperedges contains vertices of both colors. The exact problem which we deal with is an "above guarantee" parameterized version of r-Set Splitting. We prove a dichotomy on its time complexity. In particular, we show that this problem is fixed parameter tractable when $r < \log n$, but its complexity explodes when $r \geq \log n$. We also show a linear vertex-kernel when $r = O(1)$.

Finally, we return to the graph parameter of tree-depth. We study its extension on directed graphs from the scope of Graph Searching. We show a complete game theoretic characterization and a min-max theorem both for simple and for directed graphs.

1.3 The Structure of This Thesis

In Chapter 2, we define the necessary notions from Graph Theory and Logic, which we use throughout the text.

In Chapter 3, we discuss the most important partial orderings on Graphs and extensively mention the basic results of Graph Minors Theory on them. Then, inspired by Kuratowski's Theorem, we show a theorem that decomposes the graphs that do not contain the Kuratowski graphs as immersions. Finally, we discuss the main algorithmic applications of the Graph Minors Theory.

In Chapter 4, we study the parameter of tree-depth, which is closed under taking of minors. We show that the size of the obstructions, that is, the minor-minimal graphs that do not belong in the graph class, of $\mathcal{G}_k$ is bounded by a function of k of order $2^{2^{k-1}}$. We also give a precise structural characterization of the acyclic obstructions, which in turn allows us to compute exactly all the acyclic obstructions of $\mathcal{G}_k$. These are $\frac{1}{2}2^{2^{k-1}-k}(1+2^{2^{k-1}-k})$. Then, we identify all the forbidden minors for the cases where $k \in \{1, 2, 3\}$. Finally, we show a theorem that, given a graph G of tree-depth k, allows us to find proper subgraphs of G of tree-depth k.

In Chapter 5, we study the problem of the computation of obstruction sets for graph classes that are closed under taking of immersions. We show that, in order to effectively compute the obstruction set of a graph class, it is enough to compute an upper bound on the tree-width of the obstructions and a description of the graph class in Monadic Second Order Logic.

In Chapter 6, we show an optimized version of the Weak Structure Theorem of the Graph Minors Theory, in which all the parameter dependencies are optimal. Furthermore, we discuss how a corollary of this theorem and an already known theorem, that may also be obtained as a corollary of our result, "extend" the duality of the regular tiling with triangles and the regular tiling with hexagons to graphs that forbid an apex graph as a minor.

In Chapter 7, we show a structural characterization for the graphs that are embedded in a surface of Eüler genus γ and forbid a fixed graph H as an immersion. In particular, we show that in this case, the graph has either bounded tree-width (where the bound is a function of H and γ) or the edge-connectivity of the graph is bounded by the maximum degree of H.

In Chapter 8, we define the necessary notions from Parameterized Complexity Theory. We then show how we may apply the optimized version of the Weak Structure Theorem in order to obtain a proof of a result of F. Dragan, F. Fomin and P. Golovach on spanners of graphs [67] in a simpler and shorter way. We continue by discussing the basic elements of Bidimensionality Theory and how we may use them in order to show that two dual graph parameters have the Erdős-Pósa property. Finally, we discuss the Kernelization Theory and its equivalence to fixed parameter tractability.

In Chapter 9, we deal with an "above guarantee" parameterization of Set Splitting. First, we show a generalization of the lower bound on the size of a cut that was given by Edwards, for the case of the partition-connected hypergraphs. Then, by combining it with two reduction rules we show a linear vertex-kernel when the size of the hyperedges is $O(1)$. Subsequently, we show that the problem is fixed parameter tractable when the size of the hyperedges is less than $\log n$, where n is the number of the vertices. We complement this result by showing that the problem is not expected to belong to XP, that is, a solution cannot be computed in $n^{f(k)}$ steps, when the size of the hyperedges is at least $\log n$.

Finally, in Chapter 10, we give an alternative definition of the graph parameter of tree-depth and for the graph parameter of cycle-rank, which is an extension of tree-depth in digraphs, in terms of Graph Searching. We show that the games are monotone both in the simple and in the directed case and the number of necessary searchers does not depend on the visibility of the fugitive. Finally, we show min-max theorems for these parameters which reveal the structure of their obstructions.

Part of the results of this thesis has been presented in the papers [69, 97–105].

1.4 The Papers

[69] Zdenek Dvorak, Archontia C. Giannopoulou, and Dimitrios M. Thilikos. Forbidden graphs for tree-depth. *Eur. J. Comb.*, 33(5):969–979, 2012.

This publication is based on parts of Chapter 4.

[97] Archontia C. Giannopoulou, Paul Hunter, and Dimitrios M. Thilikos. Lifo-search: A min-max theorem and a searching game for cycle-rank and tree-depth. *Discrete Applied Mathematics*, 160(15):2089–2097, 2012.

This publication is based on parts of Chapter 10.

[98] Archontia C. Giannopoulou, Marcin Kamiński, and Dimitrios M. Thilikos. Excluding graphs as immersions in surface embedded graphs. unpublished manuscript.

This paper is based on Chapter 7.

[99] Archontia C. Giannopoulou, Marcin Kamiński, and Dimitrios M. Thilikos. Forbidding kuratowski graphs as immersions. *CoRR*, abs/1207.5329, 2012.

This paper is based on a part of Chapter 3.

[100] Archontia C. Giannopoulou, Sudeshna Kolay, and Saket Saurabh. New lower bound on max cut of hypergraphs with an application to r -set splitting. In David Fernández-Baca, editor, *LATIN*, volume 7256 of *Lecture Notes in Computer Science*, pages 408–419. Springer, 2012.

This publication is based on Chapter 9.

[101] Archontia C. Giannopoulou, Iosif Salem, and Dimitris Zoros. Effective computation of immersion obstructions for unions of graph classes. In Fedor V. Fomin and Petteri Kaski, editors, *SWAT*, volume 7357 of *Lecture Notes in Computer Science*, pages 165–176. Springer, 2012.

This publication is based on parts of Chapter 5.

[102] Archontia C. Giannopoulou, Iosif Salem, and Dimitris Zoros. Effective computation of immersion obstructions for unions of graph classes. *CoRR*, abs/1207.5636, 2012.

This paper is a complete form of [101].

[103] Archontia C. Giannopoulou and Dimitrios M. Thilikos. Obstructions for tree-depth. *Electronic Notes in Discrete Mathematics*, 34:249–253, 2009.

This publication is a preliminary version of [69].

[104] Archontia C. Giannopoulou and Dimitrios M. Thilikos. A min-max theorem for lifo-search. *Electronic Notes in Discrete Mathematics*, 38:395–400, 2011.

This publication is a preliminary version of [97].

[105] Archontia C. Giannopoulou and Dimitrios M. Thilikos. Optimizing the graph minors weak structure theorem. *CoRR*, abs/1102.5762, 2011.

This paper is based on parts of Chapter 6.

CHAPTER 2

Basic Notions

Let $n \in \mathbb{N}$. We denote by $[n]$ the set $\{1, 2, \ldots, n\}$. Moreover, for every $k \leq n$, if S is a set such that $|S| = n$, we say that a set $S' \subseteq S$ is a *k-subset* of S if $|S'| = k$.

Given a set S, we denote by $\mathcal{P}(S)$ its power-set.

2.1 Graphs

Graphs, Hypergraphs, Multigraphs and Digraphs

A *graph* G is an ordered pair (V, E) where V is a finite set, called the *vertex set* and denoted by $V(G)$, and E is a set of 2-subsets of V, called the *edge set* and denoted by $E(G)$. We denote by $\mathbf{n}(G)$ the number of vertices of G, that is, $\mathbf{n}(G) = |V(G)|$, and by $\mathbf{m}(G)$ the number of its edges, that is, $\mathbf{m}(G) = |E(G)|$.

If we allow E to be a subset of $\mathcal{P}(V)$, then we call the pair $H = (V(H), E(H))$ a *hypergraph*.

Evenmore, if we allow $E(G)$ to be a *multiset*, that is, a set S may appear more than one times in $E(G)$, then the graph G is called a *multigraph*.

Finally, if $E(G)$ is a set of ordered 2-subsets of V, then G is called a *directed graph* (or *digraph*). Unless otherwise stated, we consider finite undirected graphs without loops or multiple edges.

The graph where $V(G) = E(G) = \emptyset$ is called *empty* graph.

We call two vertices v and u of a graph G *adjacent* if $\{v, u\} \in E(G)$. Moreover, for every $v \in V(G)$ we say that an edge e of G is *incident* to v if $v \in e$.

For a vertex v, we denote by $N_G(v)$ its (*open*) *neighborhood*, that is, the set of vertices which are adjacent to v. The *closed neighborhood* $N_G[v]$ of v is the set $N_G(v) \cup \{v\}$. For $U \subseteq V(G)$, we define $N_G(U) = \bigcup_{v \in U} N_G(v)$ and $N_G[U] = \bigcup_{v \in U} N_G[v]$ the open and closed neighborhood of set U, respectively. We let $\deg_G(v) = |N_G(v)|$ denote the *degree* of the vertex v in G. Moreover, $\delta(G) = \min_{v \in V(G)} \deg_G(v)$ and $\Delta(G) = \max_{v \in V(G)} \deg_G(v)$. Note that in a multigraph G, for every $v \in V(G)$, $|N_G(v)| \leq \deg_G(v)$. We may omit the index if the graph under consideration is clear from the context. A graph G is called *sub-cubic* if $\Delta(G) \leq 3$. Finally, given a vertex v of a graph G, we denote by $E_G(v)$ the edges of G incident to v.

A graph is a *path* (respectively *cycle*) if we can arrange its vertices in a linear (respectively cyclic) sequence in such a way that two vertices are adjacent if and only if they are consecutive in the sequence. The *endpoints of a path* are the first and the last vertex of the sequence. A *walk* in a graph G is a sequence of elements of $V(G)$ (where a vertex may appear more than once) such that every two consecutive vertices in the sequence are adjacent. We denote by C_n (respectively P_n) the cycle (respectively path) that has n vertices. The *length* of a path or a cycle is defined as the number of its edges. Given a path P and $v, u \in V(P)$ we denote by $P[v, u]$ its subpath with endpoints v and u.

Two graphs are called *vertex-disjoint* if they do not share common vertices and *edge-disjoint* if they do not share common edges.

Given two paths P_1 and P_2, who share a common endpoint v, we say that they are *well-arranged* if their common vertices appear in the same order in both paths, that is, if $S = V(P_1) \cap V(P_2) = \{u, v_1, v_2, \ldots, v_{|S|-1}\}$ and $uv_1v_2 \ldots v_{|S|-1}$ is the sequence according to which the vertices of S appear on P_1 then $uv_1v_2 \ldots v_{|S|-1}$ is also the sequence according to which they appear in P_2.

Given two vertices v, u in a graph G their *distance* in G, denoted by $\mathbf{dist}_G(v, u)$, is equal to the minimum length of a path in G with endpoints v and u. Evenmore, given a vertex v (respectively vertex set S) in a graph G, the *neighborhood* of v (respectively S) *at distance at most* r, denoted by $N_G^r[v]$ (respectively $N_G^r[S]$), is the set consisting of all vertices of G of distance at most r from v (respectively the vertices of S).

A graph G is *chordal* (or *triangulated*) if every cycle C, of length $k \geq 4$, contains a *chord*, where a chord is an edge joining two non-consecutive vertices of C. A *triangulation* of a graph G is a chordal graph G' such that $V(G) = V(G')$ and $E(G) \subseteq E(G')$.

A graph is called *complete* if all of its vertices are pairwise adjacent and the *complete graph* on n vertices is denoted by K_n. Moreover, if S is a finite set, we denote by $K[S]$ the complete graph with vertex set S.

A graph G is called *bipartite* if we can partition its vertex set into two subsets X and Y such that every edge has an endpoint in X and an endpoint in Y. The sets X and Y are called the *parts* of the partition. A bipartite graph G is called *complete bipartite* if for every vertex $x \in X$, $N_G(x) = Y$ and every vertex $y \in Y$, $N_G(y) = X$. We denote by $K_{q,r}$ the *complete bipartite graph* where one part has size q and the other part has size r, where $q, r \in \mathbb{N}$. In the case where $q = 1$ this graph, that is, the graph $K_{1,r}$, is also called *star*.

A graph is called *connected* if, for every partition of its vertex set into two non-empty subsets X and Y, there is an edge that has one endpoint in X and one endpoint in Y. A graph is called a *tree* if it is connected and no subset of its vertices and its edges forms a cycle. Given a tree T, we call *leaves* its vertices of degree 1 and say that the tree is *ternary* if all of its vertices that are not leaves have degree exactly 3.

Given two graphs G and G', their *union* $G \cup G'$ and their *intersection* $G \cap G'$ are the graphs with $V(G \cup G') = V(G) \cup V(G')$ and $E(G \cup G') = E(G) \cup E(G')$ and, $V(G \cap G') = V(G) \cap V(G')$ and $E(G \cap G') = E(G) \cap E(G')$, respectively. If $G \cap G' = \emptyset$ then the graphs G and G' are called *disjoint* and their union is called *disjoint union*. Let $\mathcal{C}$ be a class of graphs and S be a set of vertices. We denote by $\pmb{\bigcup}\mathcal{C}$ the graph $\cup_{G \in \mathcal{C}} G$ and we set $\mathcal{C} \setminus S = \{G \setminus S \mid G \in \mathcal{C}\}$.

It is easy to see that, every graph G can be uniquely expressed as a disjoint union of connected graphs. These graphs are called *connected components* of G. Let $\mathcal{C}(G)$ denote the connected components of a graph G. When the connected components of a graph are trees, then the graph is called *forest*.

The *line graph* of a graph G, denoted by $L(G)$, is the graph $(E(G), X)$, where $X = \{\{e_1, e_2\} \subseteq E(G) \mid e_1 \cap e_2 \neq \emptyset \wedge e_1 \neq e_2\}$.

Given two graphs G and H, the *lexicographic product* $G \times H$, is the graph with $V(G \times H) = V(G) \times V(H)$ and $E(G \times H) = \{\{(x, y), (x', y')\} \mid (\{x, x'\} \in E(G)) \vee (x = x' \wedge \{y, y'\} \in E(H))\}$.

Evermore, the *cartesian product* of G and H is the graph $G * H$, with $V(G * H) = V(G) \times V(H)$ and $E(G * H) = \{\{(x, y), (x', y')\} \mid (\{x, x'\} \in E(G) \wedge y = y') \vee (x = x' \wedge \{y, y'\} \in E(H))\}$.

A *k-coloring* of a graph G is a function $f : V(G) \to [k]$ assigning *colors* to its vertices. A k-coloring is *proper* if no two adjacent vertices are assigned the same color. The *chromatic number* of a graph G is the minimum integer k for which G admits a proper k-coloring.

Graph Operations

Let G be a graph, $v \in V(G)$ and $e \in E(G)$. The graph $G - v$, obtained from G after the removal of v is the graph where $V(G - u) = V(G) \setminus \{v\}$ and $E(G - v) = \{e \in E(G) \mid e \cap \{v\} = \emptyset\}$. The graph $G - e$, obtained from G after the removal of e is the graph where $V(G - e) = V(G)$ and $E(G-e) = E(G) \setminus \{e\}$. If $U \subseteq V(G)$ ($E \subseteq E(G)$) then $G-U$ (respectively $G - E$) is the graph obtained from G after the removal of the vertices of U (respectively of the edges of E).

Given an edge $e = \{x, y\}$ of a graph G, the graph G/e is obtained from G by *contracting* the edge e, that is, the endpoints x and y are replaced by a new vertex v_{xy} which is adjacent to the old neighbors of x and y (except x and y). Furthermore, if one of the endpoints of e, say v has degree 2 then the contraction of e is also called *dissolution of the vertex v*.

The *lift* of two edges $e_1 = \{x, y\}$ and $e_2 = \{x, z\}$ to an edge e is the operation of removing e_1 and e_2 from G and then adding the edge $e = \{y, z\}$ in the resulting graph.

Let G be a graph such that $K_3 \subseteq G$ and x, y, z be the vertices of K_3. The *ΔY-transformation of K_3 in G* is the following: We remove the edges $\{x, y\}, \{y, z\}, \{x, z\}$, add a new vertex w, and then add the edges $\{x, w\}, \{y, w\}, \{z, w\}$.

Connectivity

Let k be an integer. A graph G is *k-connected* if, for every $S \subseteq V(G)$ with $|S| < k$, the graph $G \setminus S$ is connected. The *connectivity* of a graph G is the maximum integer k for which G is k-connected. A set $S \subseteq V(G)$ for which $G \setminus S$ is not connected is called a *separator* of G.

Similarly, a graph G is *k-edge-connected* if, for every $F \subseteq V(G)$ with $|F| < k$, the graph $G \setminus F$ is connected. The *edge-connectivity* of a graph G is the maximum integer k for which G is k-edge-connected. An *edge-cut* in a graph G is a non-empty set F of edges that belong to the same connected component of G and such that $G \setminus F$ has more connected components than G. If $G \setminus F$ has one more connected component than G then we say that F is a *minimal* edge-cut. Let F be an edge cut of a graph G and let G' be the connected component of G containing the edges of F. We say that F is an *internal edge-cut* if it is minimal and both connected components of $G' \setminus F$ contain at least 2 vertices. Finally, if $|F| = k$, we call F a *k-edge-cut*.

Partial orderings on graphs

For a graph H we say that it is

- an *induced subgraph* of a graph G, denoted by $H \sqsubseteq G$, if it can be obtained from G by applying vertex deletions.

- an *spanning subgraph* of a graph G, denoted by $H \subseteq_s G$, if it can be obtained from G by applying edge deletions.

- a *subgraph* of a graph G, denoted by $H \subseteq G$, if it can be obtained from G by applying edge and vertex deletions.

- a *contraction* of a graph G, denoted by $H \leq_c G$, if it can be obtained from G be applying edge contractions. An alternative, and sometimes more useful for our purposes, definition of a contraction is the following.

Let G and H be graphs and let $\phi : V(G) \rightarrow V(H)$ be a surjective mapping such that

1. for every vertex $v \in V(H)$, its codomain $\phi^{-1}(v)$ induces connected graph $G[\phi^{-1}(v)]$;
2. for every edge $\{v,u\} \in E(H)$, the graph $G[\phi^{-1}(v) \cup \phi^{-1}(u)]$ is connected;
3. for every $\{v,u\} \in E(G)$, either $\phi(v) = \phi(u)$, or $\{\phi(v), \phi(u)\} \in E(H)$.

We then say that H *is a contraction of* G *via* ϕ and denote it by $H \leq_c^{\phi} G$. If $H \leq_c^{\phi} G$ and $v \in V(H)$, then we call the codomain $\phi^{-1}(v)$ the *model of* v in G.

- a *minor* of a graph G, denoted by $H{\leq_m}G$, if it can be obtained from G by applying edge and vertex deletions and edge contractions.

  Alternatively, a graph H is a minor of G, if there is a function that maps every vertex v of H to a connected set $B_v \subseteq V(G)$, such that for every two distinct vertices v, w of H, B_v and B_w share no common vertex, and for every edge $\{u,v\}$ of H, there is an edge in G with one endpoint in B_v and one in B_u.

  The graph that is obtained by the union of all B_v such that $v \in V(H)$ and by the edges between B_v and B_u in G, if there exists an edge $\{v,u\}$ in H, is called a *model of* H *in* G. A model with minimal number of vertices and edges is called *minimal model.*

  We say that a graph G is H*-minor-free* if it does not contain H

as a minor. We also say that a graph class $\mathcal{G}$ is *H-minor-free* (or, excludes H as a minor) if all its members are H-minor-free.

- a *topological minor* of a graph G, denoted by $H \leq_{tm} G$, if it can be obtained from G by applying vertex deletions, edge deletions, and vertex dissolutions. Moreover, we say that a graph G is a *subdivision* of a graph H, if H can be obtained from G by dissolving vertices.

- an *immersion* of a graph G, denoted by $H{\leq_{im}}G$, if it can be obtained from G by applying vertex deletions, edge deletions, and edge lifts.

  Equivalently, we say that H is an immersion of G if there is an injective mapping $f : V(H) \to V(G)$ such that, for every edge $\{u, v\}$ of H, there is a path from $f(u)$ to $f(v)$ in G and for any two distinct edges of H the corresponding paths in G are edge-disjoint.

  Additionally, if these paths are internally disjoint from $f(V(H))$, then we say that H is *strongly immersed* in G.

  As above, the function f is called a *model of H in G* and a model with minimal number of vertices and edges is called *minimal model*.

Tree-width and Linkages

A *tree decomposition* of a graph G is a pair $(\mathcal{X}, T)$, where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of $V(G)$, called *bags* such that:

1. $\bigcup_{i \in V(T)} X_i = V(G)$;

2. for each edge $\{x, y\} \in E(G)$, $\{x, y\} \subseteq X_i$ for some $i \in V(T)$, and

3. for each $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is

$$\max_{i \in V(T)} \{|X_i| - 1\}.$$

The *tree-width* of a graph G is the minimum width over all tree decompositions of G. (In the case where T is a path, then the decomposition above is called *path decomposition* and its *width* as well as the *path-width* of a graph are defined similarly.)

Let r be a positive integer. An *r-approximate linkage* in a graph G is a family L of paths with distinct endpoints in G such that for every $r+1$ distinct paths $P_1, P_2, \dots, P_{r+1}$ in L, it holds that $\bigcap_{i \in [r+1]} V(P_i) = \emptyset$. We call these paths the *components* of the linkage. Let $(\alpha_1, \alpha_2, \dots, \alpha_k)$ and $(\beta_1, \beta_2, \dots, \beta_k)$ be elements of $V(G)^k$. We say that an r-approximate linkage L, consisting of the paths $P_1, P_2, \dots, P_k$, *links* $(\alpha_1, \alpha_2, \dots, \alpha_k)$ and $(\beta_1, \beta_2, \dots, \beta_k)$ if P_i is a path with endpoints α_i and β_i, for every $i \in [k]$. The *order* of such linkage is k. We call an r-approximate linkage of order k, *r-approximate k-linkage.*

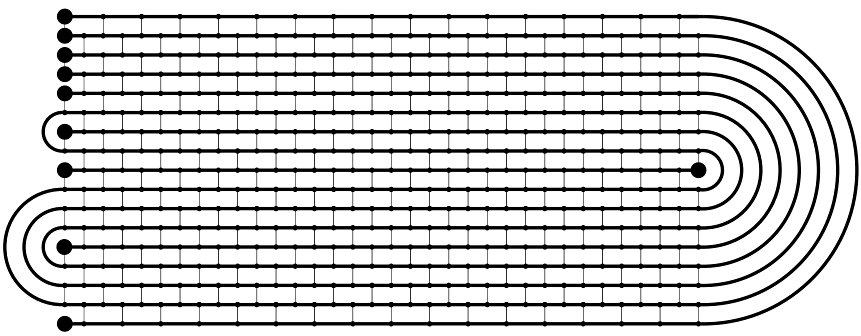


Figure 2.1: A unique linkage in a graph.

Two r-approximate k-linkages L and L' are equivalent if they have the same order and for every component P of L there exists a component P' of L' with the same endpoints. An r-approximate linkage L of a graph G is called *unique* if for every equivalent linkage L' of L, $V(L) = V(L')$. When $r = 1$, such a family of paths is called *linkage*. Finally, a linkage L in a graph G is called *vital* if $V(L) = V(G)$ and there is no other linkage in G linking the same pairs of vertices. (For an example of a unique linkage, see Figure 2.1.)

2.2 Graphs on Surfaces

A graph which can be drawn in the plane in such a way that the edges meet only at points corresponding to their common ends is called a *planar graph*. Such a drawing of a planar graph is called a *planar embedding* of the graph and a planar graph together with its embedding is called a *plane graph*. Notice that if G is a plane graph then $\mathbb{R}^2 \setminus G$ is open. The connected components of $\mathbb{R}^2 \setminus G$ are called the *faces* of G and the closed walks, also called *tours*, that bound the faces are called *facial cycles*.

Let G be a graph. We say that G is an *apex graph* if there exists a vertex $v \in V(G)$ such that $G \setminus v$ is planar. Moreover, we say that G is an *α-apex graph* if there exists a set $S \subseteq V(G)$ such that $|S| \leq \alpha$ and $G \setminus S$ is planar. We denote by $\mathbf{an}(G)$, the minimum $k \in \mathbb{N}$ such that G is a k-apex graph, that is,

$$\mathbf{an}(G) = \min\{k \in \mathbb{N} \mid \exists S \subseteq V(G) : (|S| \leq k \wedge G \setminus S \text{ is planar.})\}$$

Clearly, $\mathcal{G} = \{G \mid \mathbf{an}(G) = 1\}$ is the class consisting of all apex graphs.

We say that a hypergraph H is *planar* if its incidence graph is planar, where the *incidence graph* of a hypergraph H is the bipartite graph $I(H)$

on the vertex set $V(H) \cup E(H)$ where $v \in V(H)$ is adjacent to $e \in E(H)$ if and only if $v \in e$, that is, v is incident to e in H.

A *surface* Σ is a compact 2-manifold without boundary (we always consider connected surfaces). If, instead of the plane, we consider a surface Σ, we say that a graph G is Σ*-embeddable*, if it can be drawn in Σ in such a way that its edges meet only at their common endpoints. Whenever we refer to a Σ-embedded graph G we consider a 2-cell embedding of G in Σ, where a *2-cell embedding* of a graph G is an embedding in which every face is homeomorphic to an open disk. To simplify notations we do not distinguish between a vertex of G and the point of Σ used in the drawing to represent the vertex or between an edge and the arc representing it. We also consider a graph G embedded in Σ as the union of the points corresponding to its vertices and edges. That way, a subgraph H of G can be seen as a graph H, where $H \subseteq G$ in Σ. Recall that $\Delta \subseteq \Sigma$ is an open (respectively closed) disc if it is homeomorphic to $\{(x,y) : x^2 + y^2 < 1\}$ (respectively $\{(x,y) : x^2 + y^2 \leq 1\}$). The *Eüler genus* of a non-orientable surface Σ is equal to the non-orientable genus $\tilde{\mathbf{g}}(\Sigma)$ (or the crosscap number). The *Eüler genus* of an orientable surface Σ is $2\mathbf{g}(\Sigma)$, where $\mathbf{g}(\Sigma)$ is the orientable genus of Σ. We refer to the book of Mohar and Thomassen [157] for more details on graphs embeddings. The *Eüler genus* of a graph G (denoted by $\mathbf{eg}(G)$) is the minimum integer γ such that G can be embedded on a surface of the Euler genus γ.

Notice that the graphs that are embedded in the sphere $\mathbb{S}_0$ are the planar graphs. Evenmore, we call such a graph, along with its embedding, Σ_0*-embedded graph*. Let C_1, C_2 be two disjoint cycles in a Σ_0-embedded graph G. Let also Δ_i be the open disk of $\mathbb{S}_0 \backslash C_i$ that does not contain points of C_{3-i}, $i \in [2]$. The *annulus between* C_1 and C_2 is the set $\mathbb{S}_0 \setminus (\Delta_1 \cup \Delta_2)$ and we denote it by $A[C_1, C_2]$. Notice that $A[C_1, C_2]$ is a closed set. If

$\mathcal{A} = \{C_1, \ldots, C_r\}$ is a collection of cycles of a $\mathbb{S}_0$-embedded graph G. We say that $\mathcal{A}$ is *nested* if, for every $i \in [r-2]$, $A[C_i, C_{i+1}] \cup A[C_{i+1}, C_{i+2}] = A[C_i, C_{i+2}]$. (See, for example, Figure 2.2)

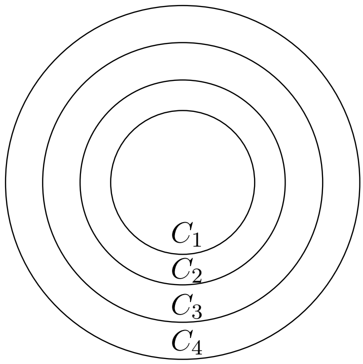


Figure 2.2: A family of four nested cycles.

Let G be a graph embedded in some surface Σ and let $x \in V(G)$. We define a *disk around* x as any open disk Δ_x with the property that each point in $\Delta_x \cap G$ is either x or belongs to the edges incident to x. Let P_1 and P_2 be two edge-disjoint paths in G. We say that P_1 and P_2 are *confluent* if, for every $x \in V(P_1) \cap V(P_2)$, that is not an endpoint of P_1 or P_2, and for every disk Δ_x around x, one of the connected components of the set $\Delta_x \setminus P_1$ does not contain any point of P_2. We also say that a collection of paths is *confluent* if the paths in it are pairwise confluent.

Moreover, given two edge-disjoint paths P_1 and P_2 in G, we say that a vertex $x \in V(P_1) \cap V(P_2)$ that is not an endpoint of P_1 or P_2 is an *overlapping vertex of* P_1 *and* P_2 if there exists a Δ_x around x such that both connected components of $\Delta_x \setminus P_1$ contain points of P_2. (See, for example, Figure 2.3) For a family of paths $\mathcal{P}$, a vertex v of a path $P \in \mathcal{P}$ is called an *overlapping vertex of* P if there exists a path $P' \in \mathcal{P}$, with

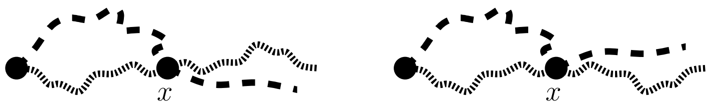


Figure 2.3: Vertex x is an overlapping vertex of the paths on the left, but it is not an overlapping vertex of the paths on the right.

$P' \neq P$, such that v is an overlapping vertex of P and P'.

2.3 Grids and Walls

Let k and r be positive integers where $k, r \geq 2$. The *$(k \times r)$-grid* is the Cartesian product of two paths of lengths $k-1$ and $r-1$ respectively. A vertex of a $(k \times r)$-grid is a *corner* if it has degree 2. Thus each $(k \times r)$-grid has 4 corners. A vertex of a $(k \times r)$-grid is called *internal* if it has degree 4, otherwise it is called *external*. (For an example of a grid, see Figure 2.4.)

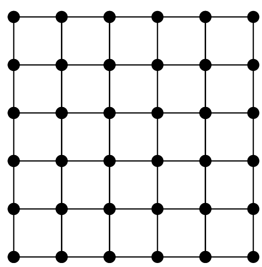


Figure 2.4: The (6×6)-grid.

A *wall of height k*, $k \geq 1$, is the graph obtained from a $((k+1) \times (2 \cdot k+2))$-grid with vertices (x, y), $x \in \{1, \ldots, 2 \cdot k+4\}$, $y \in \{1, \ldots, k+1\}$, after the removal of the "vertical" edges $\{(x, y), (x, y+1)\}$ for odd $x+y$, and then the removal of all vertices of degree 1. We denote such a wall by W_k.

The *corners* of the wall W_k are the vertices $c_1 = (1,1)$, $c_2 = (2\cdot k+1, 0)$, $c_3 = (2\cdot k+1+(k+1 \mod 2), k+1)$ and $c_4 = (1+(k+1 \mod 2), k+1)$. (The square vertices in Figure 2.5.) We let $C = \{c_1, c_2, c_3, c_4\}$ and we call the pairs $\{c_1, c_3\}$ and $\{c_2, c_4\}$ *anti-diametrical.*

A *subdivided wall W of height k* is a wall obtained from W_k after replacing some of its edges by paths without common internal vertices. We call the resulting graph W a *subdivision* of W_k and the vertices that appear in the wall after the replacement *subdivision vertices.*

The non-subdivision vertices of W are called *original vertices.* The *perimeter P* of a subdivided wall is the cycle defined by its boundary.

The *layers* of a subdivided wall W of height k are recursively defined as follows. The first layer of W is its perimeter. For $i = 2, \cdots, \lceil \frac{k}{2} \rceil$, the i-th layer of W is the $(i-1)$-th layer of the subwall W' obtained from W after removing from W its perimeter and all occurring vertices of degree 1 (see Figure 2.5).

If W is a subdivided wall of height k, we call *brick* of W any facial cycle whose non-subdivided counterpart in W_h has length 6. We say that two bricks are *neighbors* if their intersection contains an edge.

Let W_k be a wall. We denote by $P_j^{(h)}$ the shortest path connecting the vertices $(1, j)$ and $(2\cdot k+2, j)$ and call these paths the *horizontal paths of* W_k. Note that these paths are vertex-disjoint. We call the paths $P_{k+1}^{(h)}$ and $P_1^{(h)}$ the *southern path of* W_k and *northern path of* W_k respectively. Similarly, we denote by $P_i^{(v)}$ the shortest path connecting vertices $(i, 1)$ and $(i, k+1)$ with the assumption that for, $i < 2\cdot k+2$, $P_i^{(v)}$ contains only vertices (x, y) with $x = i, i+1$. Notice that there exists a unique subfamily $\mathcal{P}_v$ of $\{P_i^{(v)} \mid i < 2\cdot k+2\}$ of $k+1$ vertical paths with one

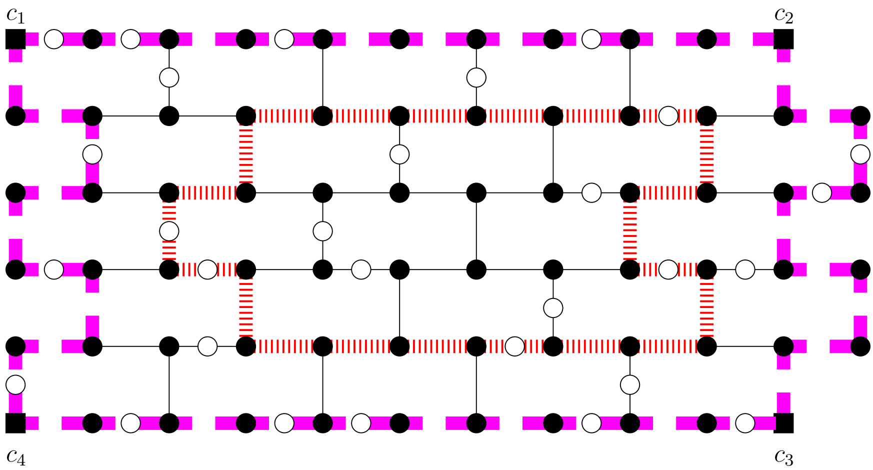


Figure 2.5: The first (magenta) and second (red) layers of a wall of height 5.

endpoint in the southern path of W_k and one in the northern path of W_k. We call these paths *vertical paths of* W_k and denote them by $P_i^{[v]}$, $i \in [k]$, where $P_1^{(v)} = P_1^{[v]}$ and $P_{2\cdot k+1}^{(v)} = P_{k+1}^{[v]}$. (See Figure 2.6.)

The paths $P_1^{[v]}$ and $P_{k+1}^{[v]}$ are called the *western path of* W_k and the *eastern path of* W_k respectively. Notice that each vertex $u \in V(W_k) \setminus V(P)$, is contained in exactly one vertical path, denoted by $P_u^{(v)}$, and in exactly one horizontal path, denoted by $P_u^{(h)}$, of W_k. If W is a subdivision of W_k, we will use the same notation for the paths obtained by the subdivisions of the corresponding paths of W_k, where we assume that u is an original vertex of W.

Let W be a wall and K' be the connected component of $G \setminus P$ that contains $W \setminus P$, where P is the perimeter of W. The *compass* K of W in

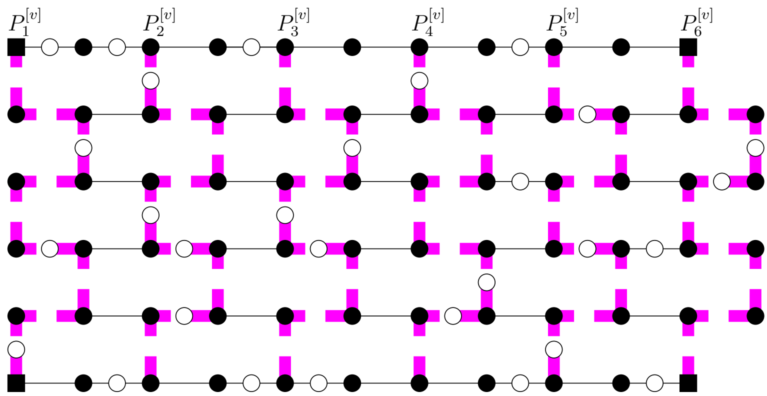


Figure 2.6: The vertical paths of a wall of height 5.

G is the graph $G[V(K') \cup V(P)]$. Observe that W is a subgraph of K and K is connected.

2.4 Logic

A very important area of both Mathematics and Computer Science is Logic. It has been widely explored, since Aristotle was the first to suggest a formal system that was then used by Euclid. Although Logic and its history are a very interesting subject, its exploration and presentation is out of the purpose of this brief introduction. We will, however, mention R. Dedekind (1831 - 1916), G. Peano (1858 - 1932), D. Hilbert (1862 - 1943) and K. Gödel whose contribution was determining and their results and suggestions were ahead of their time.

Here, we consider logics over graphs. (For more on Logic see [73] and [156].)

First Order Logic over graphs

Definition 2.1. The syntax of the *first-order logic* is the following:

- Infinite supply of *individual variables*, usually denoted by lowercase letters x, y, z.

- First-order formulas in the language of graphs are built up from *atomic formulas* $E(x, y)$ and $x = y$ by using the usual *Boolean connectives* $\neg$ (negation), $\wedge$ (conjuction), $\vee$ (disjunction), $\rightarrow$ (implication), $\leftrightarrow$ (bi-implication), existential quantification $\exists x$ and universal quantification $\forall x$ over individual variables.

Individual variables range over vertices of a graph. The atomic formula $E(x, y)$ express adjacency, and the formula $x = y$ expresses equality. From this, the free variables, the sentences and the semantics of first-order logic are defined in the obvious way.

For example, a *dominating set* in a graph $G = (V, E)$ is a set $S \subseteq V$ such that for every $v \in V$, either v belongs to S or v is adjacent to a vertex u that belongs to S. (see Figure 2.7) The following first-order sentence (parameterized by k) $\mathbf{dom}_k$ says that a graph has a dominating set of size k:

$$\exists x_1 \exists x_2 \ldots \exists x_k \left(\bigwedge_{1 \leq i < j \leq k} \neg(x_i = x_j) \wedge \forall y \left(\bigvee_{i=1}^{k} ((y = x_i) \vee E(y, x_i)) \right) \right)$$

Monadic Second Order Logic over Graphs

We call *signature* $\tau = \{R_1, \ldots, R_n\}$ any finite set of relation symbols R_i of any (finite) arity denoted by $\mathbf{ar}(R_i)$. For the language of graphs $\mathcal{G}$ we

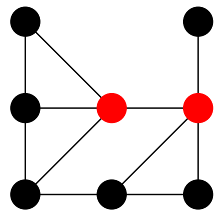


Figure 2.7: The dominating set of a graph (red vertices).

consider the signature $\tau_{\mathcal{G}} = \{V, E, I\}$ where V represents the set of vertices of a graph G, E the set of edges, and $I = \{(v, e) \,|\, v \in e \text{ and } e \in E(G)\}$ the incidence relation.

A *τ-structure* $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \ldots, R_n^{\mathfrak{A}})$ consists of a finite universe A, and the interpretation of the relation symbols R_i of τ in A, that is, for every i, $R_i^{\mathfrak{A}}$ is a subset of $A^{\mathbf{ar}(R_i)}$.

In MSO formulas are defined recursively from atomic formulas, that is, from expressions of the form $R_i(x_1, x_2, \ldots, x_{\mathbf{ar}(R_i)})$ or of the form $x = y$ where x_j, $j \leq \mathbf{ar}(R_i)$, x and y are variables, by using the Boolean connectives $\neg, \wedge, \vee, \rightarrow$, and existential or universal quantification over individual variables and sets of variables.

Notice that, in the language of graphs, the atomic formulas are of the form $V(u), E(e)$ and $I(u, e)$, where u and e are vertex and edge variables respectively. Furthermore, quantification takes place over vertex or edge variables or vertex-set or edge-set variables.

A *graph structure* $\mathfrak{G} = (V(G) \cup E(G), V^{\mathfrak{G}}, E^{\mathfrak{G}}, I^{\mathfrak{G}})$ is a $\tau_{\mathcal{G}}$-structure, which represents the graph $G = (V, E)$. From now on, we abuse notation by treating G and $\mathfrak{G}$ equally.

A graph class $\mathcal{C}$ is *MSO-definable* if there exists an MSO formula $\phi_{\mathcal{C}}$ in the language of graphs such that $G \in \mathcal{C}$ if and only if $G \models \phi_{\mathcal{C}}$, that is,

$\phi_{\mathcal{C}}$ is true in the graph G (G is a model of $\phi_{\mathcal{C}}$).

CHAPTER 3

Partial (Well-Quasi-)Orderings and Algorithms

3.1 Partial (Well-Quasi-)Orderings

One of the most famous graph-theoretic results is the characterization of planar graphs in terms of forbidden topological minors, proven by K. Kuratowski in 1930.

Theorem 3.1 (Kuratowski's Theorem [142]). *A graph G is planar if and only if it does not contain K_5 and $K_{3,3}$ as topological minors.*[1]

After the first appearance of Kuratowski's proof in [142] numerous alternative proofs were also proposed for this theorem (see, for example, [62, 63, 153]).

[1] Allegedly, this theorem had also been proved in a manuscript of L. Pontryagin and sometimes is also appears as Kuratowski - Pontryagin Theorem.

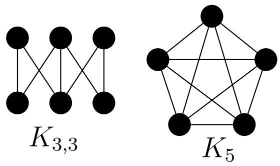


Figure 3.1: The Kuratowski graphs.

However, it was K. Wagner's restatement and proof in terms of minors in 1937 [221], and a consequent conjecture, known as Wagner's Conjecture[2], that paved the way towards studying the minor ordering on graphs and what ended up to be the enterprise now known as the Graph Minors Theory.

Theorem 3.2 ([221]). *A graph G is planar if and only if it does not contain K_5 and $K_{3,3}$ as minors.*

Let us now move into the necessary details. Notice that the two aforementioned orderings on graphs, namely the minor and the topological minor ordering, along with all other orderings that we have encountered up to now, that is, the (induced/spanning) subgraph, the contraction, and the immersion ordering, are partial orderings on graphs.

We call a family of graphs $\mathcal{F}$ an *anti-chain* for the partial ordering $\leq$ if for every two graphs $G, H \in \mathcal{F}$ neither $H \leq G$ nor $G \leq H$, that is, if the elements of $\mathcal{F}$ are mutually non-comparable with respect to the partial ordering $\leq$. Formally,

Definition 3.1. A reflexive and transitive relation is called a *partial ordering.* A partial ordering $\leq$ on X is a *well-quasi-ordering*, and the elements

[2]According to R. Diestel, while the conjecture has been attributed to Klaus Wagner, he himself had always denied discussing it, even after its proof appeared!

of X are well-quasi-ordered by $\leq$, if

- there is no strictly infinitely decreasing sequence comprising of elements of X, and
- X does not contain an infinite anti-chain.

Observe that, in the case of finite graphs there are no strictly infinitely decreasing sequences. Thus, a partial ordering on graphs is a well-quasi-ordering if there are no infinite anti-chains of graphs with respect to it. One can easily see that the (induced/spanning) subgraph ordering is not a well-quasi-ordering simply by considering the family of all cycles $\{C_i \mid i \geq 3\}$. Moreover, the class of graphs is not well-quasi-ordered with respect to the contraction relation as the class of graphs $\{K_{2,r} \mid r \in \mathbb{N}\}$ (Figure 3.2) is an infinite anti-chain for the contraction ordering.

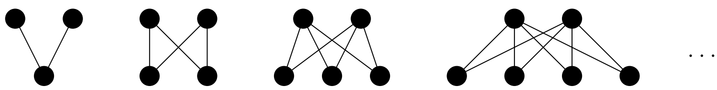


Figure 3.2: Infinite anti-chain for the contraction ordering.

Finally, in Figure 3.3 we may see some of the elements of an anti-chain for the topological minor ordering. However, as we will see later on, both the minor and the immersion ordering are well-quasi-orderings on graphs.

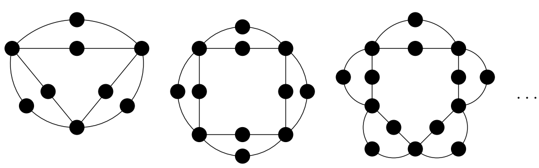


Figure 3.3: Infinite anti-chain for the topological minor ordering.

Conjecture 1 (Wagner's Conjecture)**.** *The class of graphs is well-quasi-ordered with respect to the minor ordering.*

The first successful step towards resolving Wagner's conjecture was taken by J. Kruskal [141], who proved in 1960 that the class of trees is well-quasi-ordered with respect to the minor ordering (also known as *Vázsonyi Conjecture*). In 1963 a shorter proof (again for trees) was given by Crispin St. John Alvah Nash-Williams [158]. The general case for the class of all graphs required 44 more years and a series of 23 papers, known as the Graph Minors series, until it was finally proven by N. Robertson and P. Seymour [194] and its proof classifies it as one of the deepest results of modern Combinatorics.

Theorem 3.3 (Graph Minors Theorem [194])**.** *The finite graphs are well-quasi-ordered with respect to the minor ordering.*

Evenmore, in the same series of papers, the well-quasi-ordering of the graphs with respect to immersions was proved, which was known as the *Nash-Williams' Conjecture.*

Theorem 3.4 ([211])**.** *The finite graphs are well-quasi-ordered with respect the immersion ordering.*

Given a partial ordering $\leq$, we say that a graph class $\mathcal{C}$ is *closed under taking of* this ordering if for any graph G that belongs to $\mathcal{C}$ every graph H such that $H \leq G$ also belongs to $\mathcal{C}$. For example, the class of planar graphs is closed under taking minors but it is not closed under taking immersions as K_5 is an immersion of the (7×7)-grid (Figure 3.4).

Assume now that $\mathcal{F}$ is a graph class closed under taking of minors (respectively immersions) and consider the graph class $\overline{\mathcal{F}} = \{G \mid G \notin \mathcal{F}\}$.

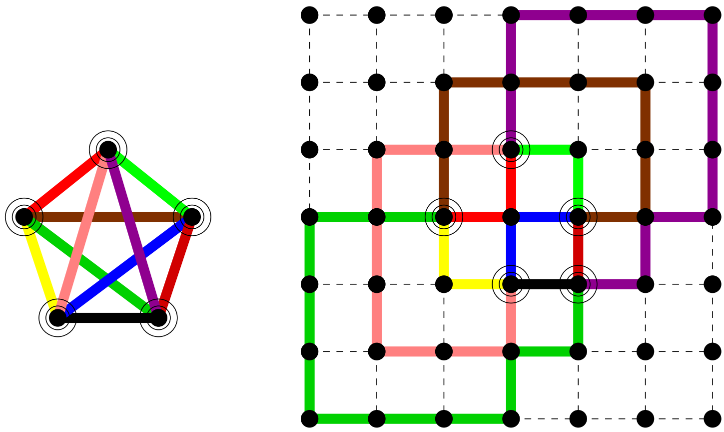


Figure 3.4: K_5 as an immersion of the (7×7)-grid.

Let $\mathcal{O}^{m}_{\overline{\mathcal{F}}}$ (respectively $\mathcal{O}^{im}_{\overline{\mathcal{F}}}$) be the set of minimal mutually non-comparable graphs in $\overline{\mathcal{F}}$ according to the minor (respectively immersion) ordering. As the minor (respectively immersion) ordering is a well-quasi-ordering $\mathcal{O}^{m}_{\overline{\mathcal{F}}}$ (respectively $\mathcal{O}^{im}_{\overline{\mathcal{F}}}$) is finite. Evenmore, a graph G belongs to $\mathcal{F}$ if and only if it does not contain any of the graphs $H \in \mathcal{O}^{m}_{\overline{\mathcal{F}}}$ (respectively $H \in \mathcal{O}^{im}_{\overline{\mathcal{F}}}$) as a minor (respectively immersion).

Indeed, assume that for some graph $G \in \mathcal{F}$ there exists a graph $H \in \mathcal{O}^{m}_{\overline{\mathcal{F}}}$ (respectively $H \in \mathcal{O}^{im}_{\overline{\mathcal{F}}}$) such that $H \leq_{\mathrm{m}} G$ (respectively $H \leq_{\mathrm{im}} G$). As $\mathcal{F}$ is closed under taking of minors (respectively immersions) and $G \in \mathcal{F}$ then $H \in \mathcal{F}$, a contradiction to the fact that $H \in \overline{\mathcal{F}}$.

For the converse, assume that there exists a graph G that does not contain any of the graphs in $\mathcal{O}^{m}_{\overline{\mathcal{F}}}$ (respectively $\mathcal{O}^{im}_{\overline{\mathcal{F}}}$) as a minor (respectively immersion) and $G \in \overline{\mathcal{F}}$. This is a contradiction to the fact that $\mathcal{O}^{m}_{\overline{\mathcal{F}}}$ (respectively $\mathcal{O}^{im}_{\overline{\mathcal{F}}}$) contains all the minor-minimal (respectively immersion-minimal) elements of $\overline{\mathcal{F}}$.

Given a graph class $\mathcal{F}$ that is closed under taking of minors (respectively immersions) we denote by $\mathbf{obs}_{\leq_{\mathrm{m}}}(\mathcal{F})$ (respectively $\mathbf{obs}_{\leq_{\mathrm{im}}}(\mathcal{F})$) the set $\mathcal{O}^{m}_{\mathcal{F}}$ (respectively $\mathcal{O}^{im}_{\mathcal{F}}$) and call it the *minor* (respectively *immersion*) *obstruction set* of F. Finally, its elements are called *minor* (respectively *immersion*) *obstructions*. For example, in the case of planar graphs, from Wagner's theorem, we know that the minor obstruction set is $\{K_5, K_{3,3}\}$, also called, the set of Kuratowski graphs.

Thus, from the above discussion one may obtain that.

Theorem 3.5. *For every graph class $\mathcal{F}$ closed under taking of minors (respectively immersions) there exists a finite set of graphs $\mathbf{obs}_{\leq_m}(\mathcal{F})$ (respectively $\mathbf{obs}_{\leq_{im}}(\mathcal{F})$) such that a graph G belongs to $\mathcal{F}$ if and only if it does not contain any of the graphs in $\mathbf{obs}_{\leq_m}(\mathcal{F})$ (respectively $\mathbf{obs}_{\leq_{im}}(\mathcal{F})$) as a minor (respectively immersion).*

3.2 Forbidding Kuratowski Graphs as Immersions

In the previous section, we saw that the Kuratowski graphs (K_5 and $K_{3,3}$) are the minor obstructions for the class of planar graphs. Evenmore, we saw that although the topological minor ordering is not a well-quasi-ordering (and thus, a graph class closed under taking topological minors is not expected to admit a characterization in terms of a finite set of forbidden topological minors), the class of planar graphs is one of the rare exceptions.

In this section, we prove a structural characterization of the graphs that exclude K_5 and $K_{3,3}$ as immersions. First of all, notice that if a graph G contains a graph H as a topological minor then it also contains it as an immersion. This directly implies that if we exclude the Kuratowski graphs as immersions we also exclude them as topological minors. Thus, a graph G that does not contain K_5 and $K_{3,3}$ as immersions is also a planar graph. In what follows, we will prove that these graphs have a special structure; they can be constructed by repetitively, joining together simpler graphs, starting from either graphs of small decomposability or from planar graphs with maximum degree 3. In particular, we prove that a graph G that does not contain neither K_5 nor $K_{3,3}$ as immersions can be constructed by applying consecutive i-edge-sums, for $i \leq 3$, to graphs that are planar sub-cubic or of branch-width at most 10.

Let us first start with some necessary definitions and preliminary results.

Edge sums. Let G_1 and G_2 be graphs, and v_1, v_2 be vertices of G_1 and G_2 respectively, such that $|E_{G_1}(v_1)| = |E_{G_2}(v_2)|$. Consider a bijection $\sigma : E_{G_1}(v_1) \to E_{G_2}(v_2)$, where $E_{G_1}(v_1) = \{e_1^i \mid i \in [k]\}$. We define the *$k$-edge sum* of G_1 and G_2 on v_1 and v_2 as the graph G obtained if we take the disjoint union of G_1 and G_2, identify v_1 with v_2, and then, for each $i \in \{1, \ldots, k\}$, lift e_1^i and $\sigma(e_1^i)$ to a new edge e^i and finally remove the vertex v_1. (See Figures 3.5 and 3.6)

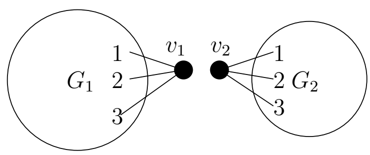


Figure 3.5: The graphs G_1 and G_2 before an edge-sum.

Let G be a graph, let F be a minimal i-edge cut in G, and let G' be the connected component of G that contains F. Let also C_1 and C_2 be the two connected components of $G' \setminus F$. We denote by C_i' the graph obtained from G' after contracting all edges of C_{3-i}' to a single vertex $v_i, i \in [2]$.

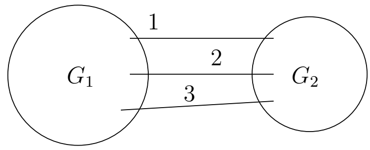


Figure 3.6: The graph obtained after an edge-sum.

We say that the graph consisting of the disjoint union of the graphs in $\mathcal{C}(G) \setminus \{C_1, C_2\} \cup \{C'_1, C'_2\}$ is the *F-split* of G and we denote it by $G|_F$. Notice that if G is connected and F is a minimal i-edge cut in G, then G is the result of an i-edge sum of the two connected components G_1 and G_2 of $\mathcal{C}(G|_F)$ on the vertices v_1 and v_2. From Menger's Theorem we obtain the following.

Observation 3.1. *Let k be a positive integer. If G is a connected graph that does not contain an internal i-edge cut, for some $i \in [k-1]$, and $v, v_1, \ldots, v_i \in V(G)$ are distinct vertices such that $\deg_G(v) \geq i$ then there exist i edge-disjoint paths from v to $v_1, v_2, \ldots, v_i$.*

Lemma 3.1. *If G is a $\{K_5, K_{3,3}\}$-immersion free connected graph and F is a minimal internal i-edge cut in G, for $i \in [3]$, then both connected components of $G|_F$ are $\{K_5, K_{3,3}\}$-immersion free.*

Proof. For contradiction assume that G is a $\{K_5, K_{3,3}\}$-immersion free connected graph and one of the connected components of $G|_F$, say C'_1, contains K_5 or $K_{3,3}$ as an immersion, where F is a minimal internal i-edge cut in G, $i \in [3]$. Assume that $H \in \{K_5, K_{3,3}\}$ is immersed in C'_1 and let $f : V(H) \to V(C'_1)$ be a model of H in C'_1. Let also v_1 be the newly introduced vertex of C'_1. Notice that if $v_1 \notin f(V(H))$ and v_1 is not an internal vertex of any of the edge-disjoint paths between the vertices in $f(V(H))$, then f is a model of H in C_1. As $C_1 \subseteq G$, f is a model of H

in G, a contradiction to the hypothesis. Thus, we may assume that either $v_1 \in f(V(H))$ or v_1 is an internal vertex in at least one of the edge-disjoint paths between the vertices in $V(H)$. Note that, as neither K_5 nor $K_{3,3}$ contain vertices of degree 1, $\deg_{C_1'}(v_1) = 2$ or $\deg_{C_1'}(v_1) = 3$.

We first exclude the case where $v_1 \notin f(V(H))$, that is, v_1 only appears as an internal vertex on the edge-disjoint paths. Observe that, as $\deg_{C_1'}(v_1) \leq 3$, v_1 belongs to exactly one path P in the model defined by f. Let v_1^1 and v_1^2 be the neighbors of v_1 in P. Recall that, by the definition of an internal F-split, there are vertices v_2^1 and v_2^2 in C_2 such that $\{v_1^1, v_2^1\}, \{v_1^2, v_2^2\} \in E(G)$. Furthermore, as C_2 is connected, there exists a (v_2^1, v_2^2)-path P' in C_2. Therefore, by substituting the subpath $P[v_1^1, v_1^2]$ with the path defined by the union of the edges $\{v_1^1, v_2^1\}, \{v_2^1, v_2^2\} \in E(G)$ and the path P' in C_2 we obtain a model of H in G defined by f, a contradiction to the hypothesis.

Thus, the only possible case is that $v_1 \in f(V(H))$. As $\delta(K_5) = 4$ and $\deg_{C_1'}(v_1) \leq 3$, f defines a model of $K_{3,3}$ in C_1'. Let v_1^1, v_1^2 and v_1^3 be the neighbors of v_1 in C_1'. We claim that there is a vertex v in C_2 and edge-disjoint paths from v to v_1^1, v_1^2, v_1^3 in G, thus proving that there exists a model of $K_{3,3}$ in G as well, a contradiction to the hypothesis. By the definition of an internal F-split, there are vertices v_2^1, v_2^2 and v_2^3 in C_2 such that $\{v_1^i, v_2^i\} \in E(G)$, $i \in [3]$. Recall that C_2 is connected. Therefore, if for every vertex $v \in C_2$, $\deg_{C_2}(v) \leq 2$, C_2 contains a path whose endpoints, say u and u' belong to $\{v_2^1, v_2^2, v_2^3\}$ and internally contains the vertex in $\{v_2^1, v_2^2, v_2^3\} \setminus \{u, u'\}$, say u''. It is easy to verify that u'' satisfies the conditions of the claim. Assume then that there is a vertex $v \in C_2$ of degree at least 3. Let G' be the graph obtained from G after removing all vertices in $V(C_1) \setminus \{v_1^1, v_1^2, v_1^3\}$ and adding a new vertex that we make it adjacent to the vertices in $\{v_1^1, v_1^2, v_1^3\}$. As G does not contain

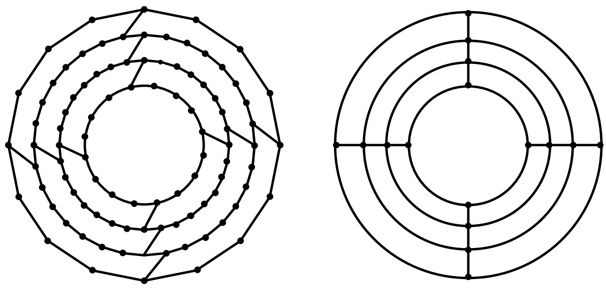


Figure 3.7: A (4,4)-railed annulus and a (4,4)-cylinder.

an internal i-edge cut, $i \in [2]$, G' does not contain an internal i-edge cut, $i \in [2]$. Therefore, from Observation 3.1 and the fact that $v \notin \{v_1^1, v_1^2, v_1^3\}$, we obtain that there exist three edge-disjoint paths from v to v_1^1, v_1^2, v_1^3 in G' and thus in G. This completes the proof of the claim and the lemma follows. $\square$

Let $r \geq 3$ and $q \geq 1$. A (r,q)-*cylinder*, denoted by $C_{r,q}$, is the Cartesian product of a cycle on r vertices and a path on q vertices. (See, for example, Figure 3.7.) A (r,q)-*railed annulus* in a graph G is a pair $(\mathcal{A}, \mathcal{W})$ such that $\mathcal{A}$ is a collection of r nested cycles $C_1, C_2, \ldots, C_r$ that are all met by a collection $\mathcal{W}$ of q paths $P_1, P_2, \ldots, P_q$ (called *rails*) in a way that the intersection of a rail and a path is always a (possibly trivial, that is, consisting of only one vertex) path. (See, for example, Figure 3.7.) Notice that given a graph G embedded in the sphere and a (k,h)-cylinder ((r,q)-railed annulus, respectively) of G, then any two cycles of the (k,h)-cylinder ((r,q)-railed annulus, respectively) define an annulus between them.

Branch decompositions. A *branch decomposition of a graph G* is a pair $\mathcal{B} = (T, \tau)$, where T is a ternary tree and $\tau : E(G) \to \mathcal{L}(T)$ is a bijection of the edges of G to the leaves of T, denoted by $\mathcal{L}(T)$. Given a branch decomposition $\mathcal{B}$, we define $\sigma_{\mathcal{B}} : E(T) \to \mathbb{N}$ as follows.

Given an edge $e \in E(T)$, let T_1 and T_2 be the trees in $T \setminus \{e\}$. Then

$$\sigma_{\mathcal{B}}(e) = |\{v \mid \text{there exist } e_i \in \tau^{-1}(\mathcal{L}(T_i)),\ i \in [2], \text{ such that } e_1 \cap e_2 = \{v\}\}|.$$

The *width of a branch decomposition* $\mathcal{B}$ is $\max_{e \in E(T)} \sigma_{\mathcal{B}}(e)$ and the *branch-width of a graph* G, denoted by $\mathbf{bw}(G)$, is the minimum width over all branch decompositions of G. When $|V(T)| \leq 1$ the width of the branch decomposition is defined to be 0. The following has been proven in [111].

Theorem 3.6 ([111])**.** *If G is a planar graph and k, h are integers with $k \geq 3$ and $h \geq 1$ then G either contains the (k,h)-cylinder as a minor or has branch-width at most $k + 2h - 2$.*

We now prove the following.

Lemma 3.2. *If G is a planar graph of branch-width at least 11, then G contains a (4,4)-railed annulus.*

Proof. Let G be a planar graph of branch-width at least 11. Then by Theorem 3.6, G contains $(4,4)$-cylinder as a minor. By the definition of the minor relation, G contains a $(4,4)$-railed annulus. □

3.2.1 (Confluent) Families of Paths

Lemma 3.3. *Let G be a graph and $v, v_1, v_2 \in V(G)$ such that there exist edge-disjoint paths P_1 and P_2 from v to v_1 and v_2 respectively. If the paths P_1 and P_2 are not well-arranged then there exist edge-disjoint paths P_1' and P_2' from v to v_1 and v_2 respectively such that $E(P_1') \cup E(P_2') \subsetneq E(P_1) \cup E(P_2)$.*

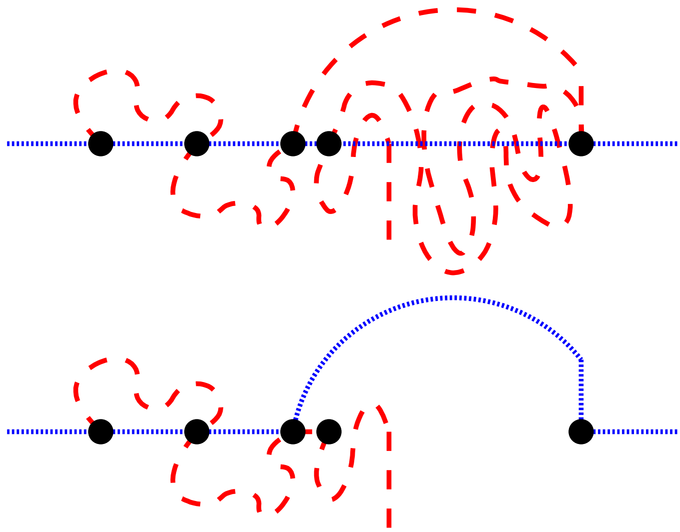


Figure 3.8: An example of the procedure in Lemma 3.3.

Proof. Let $Z = V(P_1) \cap V(P_2) = \{v, u_1, u_2, \dots, u_k\}$, where $(v,\ u_1,\ u_2, \dots, u_k)$ is the order that the vertices in Z appear in P_1, and $(v,\ u_{i_1},\ u_{i_2}, \dots,\ u_{i_k})$ is the order that they appear in P_2. As the paths are not well-arranged there exists $\lambda \in [k]$ such that $u_\lambda \neq u_{i_\lambda}$. Without loss of generality assume that λ is the smallest such integer. Also, without loss of generality, assume that $u_\lambda < u_{i_\lambda}$. We define

$$
\begin{aligned}
P_1' &= P_1[v, u_{\lambda-1}] \cup P_2[u_{\lambda-1}, u_{i_\lambda}] \cup P_1[u_{i_\lambda}, v_1] \\
P_2' &= P_2[v, u_{\lambda-1}] \cup P_1[u_{\lambda-1}, u_\lambda] \cup P_2[u_\lambda, v_2].
\end{aligned}
$$

and observe that P_1' and P_2' satisfy the desired properties. (For an example, see Figure 3.8). $\square$

Before proceeding to the statement and proof of the next proposition we need the following definition. Given a collection of paths $\mathcal{P}$ in a graph

G, we define the function $f_{\mathcal{P}} : \bigcup_{P\in\mathcal{P}} V(P) \to \mathbb{N}$ such that $f(x)$ is the number of pairs of paths $P, P' \in \mathcal{P}$ for which x is an overlapping vertex. Let

$$g(\mathcal{P}) = \sum_{x\in\bigcup_{P\in\mathcal{P}} V(P)} f_{\mathcal{P}}(x).$$

Notice that $f(x) \geq 0$ for every $x \in \bigcup_{P\in\mathcal{P}} V(P)$ and thus $g(\mathcal{P}) \geq 0$. Observe also that $g(\mathcal{P}) = 0$ if and only if $\mathcal{P}$ is a confluent collection of paths.

Lemma 3.3 allows us to prove the main result of this section. We state the result for general surfaces as the proof for this more general setting does not have any essential difference than the case where Σ is the sphere $\mathbb{S}_0$.

Proposition 3.1. *Let r be a positive integer. If G is a graph embedded in a surface Σ, $v, v_1, v_2, \ldots, v_r \in V(G)$ and $\mathcal{P}$ is a collection of r edge-disjoint paths from v to $v_1, v_2, \ldots, v_r$ in G, then G contains a confluent collection $\mathcal{P}'$ of r well-arranged edge-disjoint paths from v to $v_1, v_2, \ldots, v_r$ where $|\mathcal{P}'| = |\mathcal{P}|$ and such that $E(\bigcup_{P\in\mathcal{P}'} P) \subseteq E(\bigcup_{P\in\mathcal{P}} P)$.*

Proof. Let $\hat{G}$ be the spanning subgraph of G induced by the edges of the paths in $\mathcal{P}$ and let G' be a minimal spanning subgraph of $\hat{G}$ that contains a collection of r edge-disjoint paths from v to $v_1, v_2, \ldots, v_r$. Let also $\mathcal{P}'$ be the collection of r edge-disjoint paths from v to v_1, v_2,. . . , v_r in G' for which $g(\mathcal{P}')$ is minimum. It is enough to prove that $g(\mathcal{P}') = 0$.

For a contradiction, we assume that $g(\mathcal{P}') > 0$ and we prove that there exists a collection $\tilde{\mathcal{P}}$ of r edge-disjoint paths from v to $v_1, v_2, \ldots, v_r$ in G' such that $g(\tilde{P}) < g(\mathcal{P}')$. As $g(\mathcal{P}') > 0$, then there exists a path, say $P_1 \in \mathcal{P}'$, that contains an overlapping vertex u. Let z_1 be the endpoint of P_1 which is different from v. Without loss of generality we may assume that u is the overlapping vertex of P_1 that is closer to z_1 in P_1. Then there is a (v, z_2)-path $P_2 \in \mathcal{P}'$ such that u is an overlapping vertex of P_1 and P_2. Let $\tilde{P}_i = P_{3-i}[v, u] \cup P_i[u, z_i]$, $i \in [2]$, and $\tilde{P} = P$ for every $P \in \mathcal{P}' \setminus \{P_1, P_2\}$. As Lemma 3.3 and the edge-minimality of G' imply that the paths P_1 and P_2 are well-arranged, we obtain that $\tilde{P}_i$ is a path from v to z_i, $i \in [2]$. Let $\tilde{\mathcal{P}}$ be $\{\tilde{P} \mid P \in \mathcal{P}'\}$. It is easy to verify that $\tilde{\mathcal{P}}$ is a collection of r edge-disjoint paths from v to $v_1, v_2, \ldots, v_r$. We will now prove that $g(\tilde{\mathcal{P}}) < g(\mathcal{P}')$.

First notice that if $x \neq u$, then $f_{\tilde{\mathcal{P}}}(x) = f_{\mathcal{P}'}(x)$. Thus, it is enough to prove that $f_{\tilde{\mathcal{P}}}(u) < f_{\mathcal{P}'}(u)$. Observe that if $\{P, P'\} \subseteq \mathcal{P}' \setminus \{P_1, P_2\}$ and u is an overlapping vertex of P and P' in $\mathcal{P}'$ then u is also an overlapping vertex of $\tilde{P}$ and $\tilde{P}'$ in $\tilde{\mathcal{P}}$. Furthermore, while u is an overlapping vertex in the case where $\{P, P'\} = \{P_1, P_2\}$, it is not an overlapping vertex of $\tilde{P}_1$ and $\tilde{P}_2$. It remains to examine the case where $|\{P, P'\} \cap \{P_1, P_2\}| = 1$. In other words, we examine the case where one of the paths P and P', say P', is P_1 or P_2, and $P \in \mathcal{P}' \setminus \{P_1, P_2\}$. Let Δ_u be a disk around u and Δ_1, Δ_2 be the two distinct disks contained in the interior of Δ_u after removing P. We distinguish the following cases.

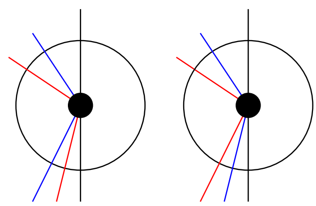


Figure 3.9: The paths P (black), P_1 (red) and P_2 (blue) and the paths $\tilde{P}_1$ (blue) and $\tilde{P}_2$ (red).

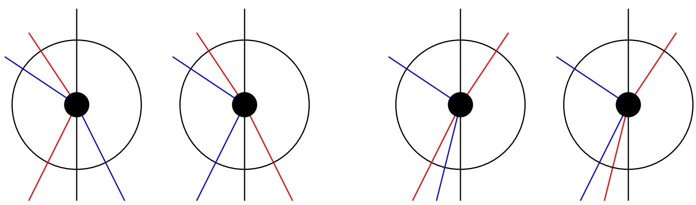
Case 1. u is neither an overlapping vertex of P_1 and P, nor of P_2 and P (see Figure 3.9). Then it is easy to see that the same holds for the pairs of paths $\tilde{P}_1$ and P, and $\tilde{P}_2$ and P. Indeed, notice that for every $i \in [2]$, P_i

intersects exactly one of Δ_1 and Δ_2. Furthermore, as u is an overlapping vertex of P_1 and P_2, both paths intersect the same disk. From the observation that $P_1 \cup P_2 = \tilde{P}_1 \cup \tilde{P}_2$, we obtain that u is neither an overlapping vertex of $\tilde{P}_1$ and P nor of $\tilde{P}_2$ and P.

Case 2. u is an overlapping vertex of P_i and P but not of P_{3-i} and P, $i \in [2]$ (see Figure 3.10). Notice that exactly one of the following holds.

- $P_i[v, u] \cup P_{3-i}[v, u]$ intersects exactly one of the disks Δ_1 or Δ_2, say Δ_1. Then $P_i[u, z_i]$ intersects Δ_2 and $P_{3-i}[u, z_{3-i}]$ intersects Δ_1. Therefore, it is easy to see that, u is not an overlapping vertex of $\tilde{P}_i$ and P but it is an overlapping vertex of $\tilde{P}_{3-i}$ and P.

- $P_i[u, z_i] \cup P_{3-i}[u, z_{3-i}]$ intersects exactly one of the disks Δ_1 or Δ_2, say Δ_1. Then $P_i[v, u]$ intersects Δ_2 and $P_{3-i}[v, u]$ intersects Δ_1. Therefore, it is easy to see that, u is an overlapping vertex of $\tilde{P}_i$ and P and is not an overlapping vertex of $\tilde{P}_{3-i}$ and P.



(α) (β)

Figure 3.10: The paths P (black), P_1 (red) and P_2 (blue) and the paths $\tilde{P}_1$ (blue) and $\tilde{P}_2$ (red).

Case 3. u is an overlapping vertex of both P_1 and P, and P_2 and P (see Figure 3.11). As above, exactly one of the following holds.

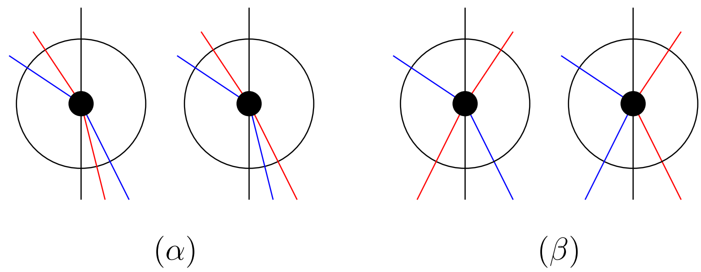


Figure 3.11: The paths P (black), P_1 (red) and P_2 (blue) and the paths $\tilde{P}_1$ (blue) and $\tilde{P}_2$ (red).

- $P_1[v,u] \cup P_2[v,u]$ intersects exactly one of the disks Δ_1 or Δ_2, say Δ_1. Then $P_1[u,z_1] \cup P_2[u,z_2]$ intersects Δ_2. It follows that u is an overlapping vertex of both $\tilde{P}_1$ and P, and $\tilde{P}_2$ and P.

- $P_1[v,u] \cup P_2[u,z_2]$ intersects exactly one of the disks Δ_1 or Δ_2, say Δ_1. Then $P_1[u,z_1] \cup P_2[v,u]$ intersects Δ_2. It follows that u is neither an overlapping vertex of $\tilde{P}_1$ and P nor of $\tilde{P}_2$ and P.

From the above cases we obtain that $f_{\tilde{\mathcal{P}}}(u) < f_{\mathcal{P}'}(u)$ and therefore $g(\tilde{\mathcal{P}}) < g(\mathcal{P}')$, contradicting the choice of $\mathcal{P}'$. This completes the proof of the proposition. $\square$

3.2.2 A Decomposition Theorem

We prove now the following decomposition theorem for $(K_5, K_{3,3})$-immersion free graphs.

Theorem 3.7. *If G is a graph not containing K_5 or $K_{3,3}$ as an immersion, then G can be constructed by applying consecutive i-edge sums, for $i \in [3]$, to graphs that either are sub-cubic or have branch-width at most 10.*

Proof. Observe first that a $(K_5, K_{3,3})$-immersion-free graph G is also $(K_5, K_{3,3})$-topological-minor-free, therefore, from Kuratowski's theorem, G is planar. Applying Lemma 3.1, we may assume that G is a $(K_5, K_{3,3})$-immersion-free graph without any internal i-edge cut, $i \in [3]$. It is now enough to prove that G is either planar sub-cubic or has branch-width at most 10. For a contradiction, we assume that $\mathbf{bw}(G) \geq 11$ and that G contains some vertex v of degree at least 4. Our aim is to prove that G contains $K_{3,3}$ as an immersion. First, let G^s be the graph obtained from G after subdividing all of its edges once. Notice that G^s contains $K_{3,3}$ as an immersion if and only if G contains $K_{3,3}$ as an immersion. Hence, from now on, we want to find $K_{3,3}$ in G^s as an immersion.

From Lemma 3.2, G and thus G^s, contains a $(4, 4)$-railed annulus as a subgraph. Observe then that G^s also contains as a subgraph a $(2, 4)$-railed annulus such that the vertex v of degree at least 4 does not belong to the annulus between its cycles. (Figure 3.12 depicts the case where v is inside the annulus between the second and the third cycle.) We denote by C_1 and C_2 the nested cycles and by R_1, R_2, R_3 and R_4 the rails of the above $(2, 4)$-railed annulus. Let A be the annulus between C_1 and C_2. Without loss of generality we may assume that C_1 separates v from C_2 and that A is edge-minimal, that is, there is no other annulus A' such that $|E(A')| < |E(A)|$ and $A' \subseteq A$.

Let now $G_1, G_2, \ldots, G_p$ be the connected components of $A \setminus (C_1 \cup C_2)$.

Claim 1. *For every $i \in [p]$ and every $j \in [2]$,*

$$|N_{G^s}(V(G_i)) \cap V(C_j)| \leq 1.$$

Proof of Claim 1. Assume the contrary. Then there is a cycle C'_j such that C'_j and $C_{j \bmod 2+1}$ define an annulus A' with $A' \subseteq A$ and $|E(A')| < |E(A)|$; a contradiction to the edge-minimality of the annulus A. □

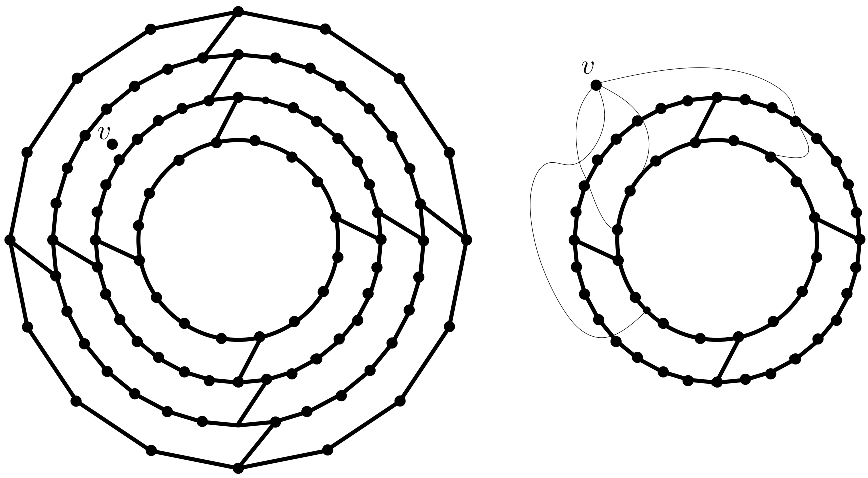


Figure 3.12: The $(4,4)$-railed annulus and the vertex v.

For every $l \in [p]$, we denote by u_1^l and u_2^l the unique neighbor of G_k in C_1 and C_2 respectively (whenever they exist). We call the connected components that have both a neighbor in C_1 and a neighbor in C_2 *substantial*. Let

$$\mathcal{C} = \{\widehat{G}_i = G[V(G_i) \cup \{u_1^i, u_2^i\}] \mid G_i \text{ is a substantial connected component}\}.$$

That is, $\mathcal{C}$ is the set of graphs induced by the substantial connected components and their neighbors in the cycles C_1 and C_2. Note that every edge of G has been subdivided in G^s and thus every edge $e \in G$ for which $e \cap C_1 \neq \emptyset$ and $e \cap C_2 \neq \emptyset$ corresponds to a substantial connected component in $\mathcal{C}$.

We now claim that there exist four confluent edge-disjoint paths P_1, P_2, P_3 and P_4 from v to C_2 in G^s. This follows from the facts that G^s does not contain an internal i-edge cut, C_2 contains at least 4 vertices, and $\deg_{G^s}(v) \geq 4$, combined with Observation 3.1. Moreover, from

Proposition 3.1, we may assume that P_1, P_2, P_3 and P_4 are confluent.

Let P_i' be the subpath $P_i[v, v_i]$ of P_i, where v_i is the vertex in $V(P_i) \cap V(C_2)$ whose distance from v in P_i is minimum, $i \in [4]$. Recall that all edges of G have been subdivided in G^s. This implies that there exist four (possibly not disjoint) graphs in $\mathcal{C}$, say $\widehat{G}_1, \widehat{G}_2, \widehat{G}_3$ and $\widehat{G}_4$ such that $v_i = u_2^i$, $i \in [4]$. We distinguish two cases.

Case 1. The graphs $\widehat{G}_1, \widehat{G}_2, \widehat{G}_3$ and $\widehat{G}_4$ are vertex-disjoint.
This implies that the endpoints of P_1', P_2', P_3' and P_4' are disjoint. Let G' be the graph induced by the cycles C_1, C_2 and the paths P_1', P_2', P_3', P_4' and let $\widehat{P}_1, \widehat{P}_2, \widehat{P}_3$ and $\widehat{P}_4$ be confluent edge-disjoint paths from v to u_2^1, u_2^2, u_2^3, and u_2^4 in G' such that

(i) $\sum\{e \mid e \in \bigcup_{i\in[4]} E(\widehat{P}_i) \setminus E(A)\}$ is minimum, that is, the number of the edges of the paths that is outside of A is minimum, and

(ii) subject to (i), $\sum\{e \mid e \in \bigcup_{i\in[4]} E(\widehat{P}_i)\}$ is minimum.

Let also $\widehat{G}$ be the graph induced by C_1, C_2, $\widehat{P}_1$, $\widehat{P}_2$, $\widehat{P}_3$, and $\widehat{P}_4$. From now on we work towards showing that $\widehat{G}$ contains $K_{3,3}$ as an immersion. For every $i \in [4]$ we call a connected component of $\widehat{P}_i \cap C_1$ non-trivial if it contains at least an edge.

Claim 2. *For every $i \in [4]$, $\widehat{P}_i \cap C_1$ contains at most one non-trivial connected component Q_i and u_1^i is an endpoint of Q_i.*

Proof of Claim 2. First, notice that any path from v to v_i in $\widehat{G}$ contains u_1^i, and thus $u_1^i \in V(\widehat{P}_i)$. Observe now that $\widehat{P}_i[u_1^i, u_2^i]$ is a subpath of $\widehat{P}_i$ whose internal vertices do not belong to C_1. Thus if u_1^i belongs to a non-trivial connected component Q_i of $\widehat{P}_i \cap C_1$, then u_1^i is an endpoint of Q_i. We will now prove that any non-trivial connected component of

$\widehat{P}_i \cap C_1$ contains u_1^i. Assume in contrary that there exists a non-trivial connected component P of $\widehat{P}_i \cap C_1$ that does not contain u_1^i. Let u be the endpoint of P for which $\mathbf{dist}_{\widehat{P}_i}(u, u_1^i)$ is minimum. Let also u' be the vertex in $\widehat{P}_i[u, u_1^i] \cap C_1$ such that $\mathbf{dist}_{\widehat{P}_i}(u, u')$ is minimum. Let P' be the subpath of C_1 with endpoints u, u' such that $\widehat{P}_i[u, u'] \cup P'$ is a cycle C with $C \cap P = \{u\}$. We further assume that the interior of $\widehat{P}_i[u, u'] \cup P'$ is the open disk that does not contain any vertices of $\widehat{P}_i$.

We will prove that for every path $\widehat{P}_j$, $j \in [4]$, $\widehat{P}_j \cap P' \subseteq \{u, u'\}$. As this trivially holds for $j = i$ we will assume that $j \neq i$. Observe that, for every $j \in [4]$, $\widehat{P}_j[v, u_1^j] \cap A \subseteq C_1$ as for every connected component H of $A \setminus (C_1 \cup C_2)$ it holds that $|N_{G^s}(V(H)) \cap V(C_j)| \leq 1$. Furthermore, observe that $\widehat{P}_i[u, u'] \cup P'$ is a separator in $\widehat{G}$. This implies that v does not belong to the interior of $\widehat{P}_i[u, u'] \cup P'$. Thus, if there is a vertex z such that $z \in \widehat{P}_j \cap (P' \setminus \{u, u'\})$, $j \neq i$, then there is a vertex $z' \in \widehat{P}_j \cap \widehat{P}_i[u, u']$, a contradiction to the confluence of the paths. We may then replace $\widehat{P}_i[u, u']$ by P', a contradiction to (i). $\square$

We denote by v_i the endpoint of Q_i that is different from u_1^i if Q_i is a non-trivial connected component of $\widehat{P}_i \cap C_1$, $i \in [4]$. Observe that $\widehat{P}_i = \widehat{P}_i[v, v_i] \cup Q_i \cup \widehat{P}_i[u_1^i, u_2^i]$, where we let $Q_i = \emptyset$ in the case where $\widehat{P}_i \cap C_1$ is edgeless, $i \in [4]$. We denote by T_i the subpath of C_1 with endpoints u_1^i and $u_1^{i \bmod 4+1}$ such that $T_i \cap \{\{u_1^1, u_1^2, u_1^3, u_1^4\} \setminus \{u_1^i, u_1^{i \bmod 4+1}\}\} = \emptyset$, $i \in [4]$. From the confluence of the paths $\widehat{P}_i$ and the fact that u_1^i is an endpoint of Q_i it follows that either $Q_i \subseteq T_i$ or $Q_i \subseteq T_{i-1}$, $i \in [4]$, where $T_{i-1} = T_{3+i \bmod 4}$ if $i - 1 \notin [4]$.

Claim 3. *There exists an $i_0 \in [4]$ such that $T_{i_0} \cap (Q_{i_0}, Q_{i_0 \bmod 4+1}) \neq T_{i_0}$.*

Proof of Claim 3. Towards a contradiction assume that for every $i \in [4]$, it holds that $T_i \cap (Q_i, Q_{i \bmod 4+1}) = T_i$. It follows that either $Q_i =$

$T_i = \widehat{P}_i[v_i, v_1^i]$, $i \in [4]$ or $Q_{i \bmod 4+1} = T_i$, $i \in [4]$. Notice then that either $v_i = u_1^{i \bmod 4+1}$, $i \in [4]$, or $v_{i \bmod 4+1} = u_1^i$, $i \in [4]$, respectively. Then, we let $\tilde{P}_{i \bmod 4+1} = \widehat{P}_i[v, v_i] \cup \widehat{P}_{i \bmod 4+1}[u_1^{i \bmod 4+1}, u_2^{i \bmod 4+1}]$ or $\tilde{P}_i = \widehat{P}_{i \bmod 4+1}[v, v_{i \bmod 4+1}] \cup \widehat{P}_i[u_1^i, u_2^i]$, $i \in [4]$, respectively. Notice that the paths $\tilde{P}_1, \tilde{P}_2, \tilde{P}_3$, and $\tilde{P}_4$ are confluent edge-disjoint paths from v to u_2^1, u_2^2, u_2^3, and u_2^4 such that $\cup_{i\in[4]}\tilde{P}_i$ is a proper subgraph of $\cup_{i\in[4]}\widehat{P}_i$. Therefore, we have that

$$\sum\{e \mid e \in \bigcup_{i\in[4]} E(\tilde{P}_i)\} < \sum\{e \mid e \in \bigcup_{i\in[4]} E(\widehat{P}_i)\},$$

a contradiction to (ii). □

It is now easy to see that $\widehat{G}$, and thus G, contains $K_{3,3}$ as an immersion. Indeed, first remove all edges of $C_1 \setminus T_{i_0}$ that do not belong to any path $\widehat{P}_i$, $i \in [4]$. Then lift the paths $\widehat{P}_i$ to a single edge where $i \neq i_0, i_0 \bmod 4+1$. Now let u_{i_0} ($u_{i_0 \bmod 4+1}$ respectively) be the vertex of T_{i_0} that belongs to $\widehat{P}_{i_0}$ ($\widehat{P}_{i_0 \bmod 4+1}$ respectively) whose distance from v in $\widehat{P}_{i_0}$ ($\widehat{P}_{i_0 \bmod 4+1}$ respectively) is minimum and lift the paths $\widehat{P}_{i_0}[v, u_{i_0}]$ and $\widehat{P}_{i_0 \bmod 4+1}[v, u_{i_0 \bmod 4+1}]$ to single edges. Notice now that $\widehat{G}$ contains the graph H_2 depicted in Figure 3.13 as an immersion. Thus, we get that $\widehat{G}$ contains $K_{3,3}$ as an immersion.

Case 2. There exist $i_1, i_2 \in [4]$ such that $\widehat{G}_{i_1}$ and $\widehat{G}_{i_2}$ are not vertex-disjoint.

Let G^μ be the graph induced by the cycles C_1 and C_2 and the graphs in $\mathcal{C}'$. We will show that G^μ contains $K_{3,3}$ as an immersion. First recall that the common vertices of $\widehat{G}_{i_1}$ and $\widehat{G}_{i_2}$ lie in at least one of the cycles C_1 and C_2. Without loss of generality assume that they have a common vertex in C_1. Recall that, as every edge of G has been subdivided in G^s, there does not

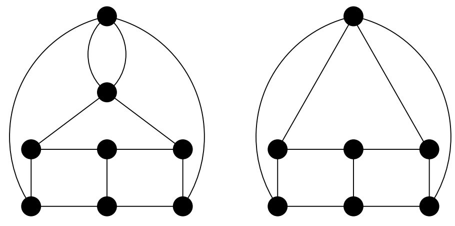


Figure 3.13: The graphs H_1 and H_2.

exist an edge $e \in G^s$ such that $e \cap C_j \neq \emptyset$, $j \in [2]$. This observation and the fact that there exist four rails between C_1 and C_2 imply that there exist at least four graphs in $\mathcal{C}'$ that are vertex-disjoint. It follows that there exist three vertex-disjoint graphs, say $\widehat{G}_{i_3}, \widehat{G}_{i_4}, \widehat{G}_{i_5}$, in $\mathcal{C}'$ with the additional properties that $\widehat{G}_{i_{2+r}} \cap \widehat{G}_{i_1} \cap C_1 = \emptyset$, $r \in [3]$, and that at most one of the $\widehat{G}_{i_3}, \widehat{G}_{i_4}, \widehat{G}_{i_5}$ has a common vertex with one of the $\widehat{G}_{i_1}, \widehat{G}_{i_2}$. Note here that none of the $\widehat{G}_{i_3}, \widehat{G}_{i_4}, \widehat{G}_{i_5}$ can have a common vertex with one of the $\widehat{G}_{i_1}, \widehat{G}_{i_2}$ in C_2, in the case where $\widehat{G}_{i_1} \cap \widehat{G}_{i_2} \cap C_2 \neq \emptyset$. It is now easy to see that G^μ contains H_1 or (H_2 respectively) depicted in Figure 3.13 as a topological minor when $\widehat{G}_{i_1} \cap \widehat{G}_{i_2} \cap C_2 \neq \emptyset$ ($\widehat{G}_{i_1} \cap \widehat{G}_{i_2} \cap C_2 = \emptyset$ respectively). Observe now that H_1 contains H_2 as an immersion. Moreover, notice that H_2 contains $K_{3,3}$ as an immersion. Thus G^μ, and therefore G^s and G, contain $K_{3,3}$ as an immersion, a contradiction. $\square$

Remark 1. It is easy to verify that our results hold for both the weak and strong immersion relations.

We believe that the upper bound on the branch-width of the building blocks of Theorem 3.7 can be further reduced, especially if we restrict ourselves to simple graphs. There is an infinite family of graphs that are not sub-cubic and have branch-width 3; two of them are depicted in

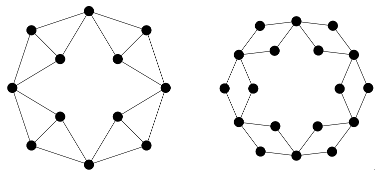


Figure 3.14: Simple non-sub-cubic graphs of branch-width 3 without K_5 or $K_{3,3}$ as immersions.

Figure 3.14. However, we have not been able to find any simple non-sub-cubic graph of branch-width greater than 3 that does not contain K_5 or $K_{3,3}$ as an immersion.

3.3 Algorithms

As we have already mentioned the proof of the Graph Minors theorem is one of the deepest in Modern Combinatorics. However, Graph minors also play an important role in the theory of Algorithms as many algorithmic techniques can be derived from the structural theorems that were proved in its context.

In this section we would like to mention, up to some extent, some of the most important (meta-)algorithms which motivated the results presented in the next chapter.

Theorem 3.8 ([193]). *Given a fixed graph H, one can construct an algorithm that decides, for any input graph G, whether $H \leq_m G$ in time $O(\mathbf{n}(G)^3)$, where the hidden constants in the O-notation depend only on H.*

By combining Theorem 3.8 with Theorem 3.5, the following is derived.

Theorem 3.9. *Given graph class $\mathcal{F}$ that is closed under taking of minors there exists an algorithm that decides, for any input graph G, whether $G \in \mathcal{F}$ in time $O(\mathbf{n}(G)^3)$, where the hidden constants in the O-notation depend only on $\mathcal{F}$.*

It is very easy to verify the correctness of the above theorem. As it is known, from Theorem 3.5, there exists a finite set of obstructions for the minor-closed graph class $\mathcal{F}$. Moreover, from Theorem 3.8, we can construct, for each one of them, an algorithm that decides whether they are contained in a graph as minors. Then, for any given graph G, we decide whether it contains any of the obstructions. If yes, then $G \notin \mathcal{F}$. Otherwise, $G \in \mathcal{F}$.

While this result on minors has been widely known since 1995, it was not until last year that the analog result was proven for the immersion ordering. In particular, in [109], M. Grohe, K. Kawarabayashi, D. Marx and P. Wollan proved the following.

Theorem 3.10. *Given a fixed graph H, one can construct an algorithm that decides, for any input graph G, whether $H \leq_{im} G$ in time $O(\mathbf{n}(G)^3)$, where the hidden constants in the O-notation depend only on H.*

As before, Theorem 3.10 combined with Theorem 3.5 yield the following.

Theorem 3.11. *Given graph class $\mathcal{F}$ that is closed under taking of immersions, there exists an algorithm that decides. for any input graph G, whether $G \in \mathcal{F}$ in time $O(\mathbf{n}(G)^3)$, where the hidden constants in the O-notation depend only on $\mathcal{F}$.*

Thus, from Theorem 3.9 (respectively Theorem 3.11) one can immediately prove that a graph class can be recognized in cubic time just by proving that it is closed under taking of minors (respectively immersions).

More generally, theorems that are providing algorithmic solution to a wide class of problems are called *meta-algorithmic theorems*. Through the past decade many such theorems have appeared [24, 36, 43, 94, 165, 168] and usually provide a generic way to identify an upper bound on the complexity of a computational problem. The most famous one is the celebrated theorem of Courcelle [36], stating that given an MSO-formula there is a linear time algorithm deciding whether an input graph G, whose tree-width is bounded, satisfies this formula.

We would like to stress here that while Theorems 3.9 and 3.11 are of great importance they crucially differ from Theorems 3.8 and 3.10. Notice that the Theorems 3.8 and 3.10 explicitly state the *construction* of an algorithm deciding whether an input graph G contains a fixed graph H as a minor or as an immersion, whereas Theorems 3.9 and 3.11 only state the *existence* of an algorithm that decides the membership of a graph into a graph class closed under taking minors or immersions.

This happens because the construction of the algorithms mentioned in the Theorems 3.9 and 3.11 assumes that the minor or immersion obstructions are known for the given graph class $\mathcal{F}$. However, the proofs in the context of the Graph Minors Theory do not provide any direction towards the identification of the obstruction sets [95]. Furthermore, this appears that it can actually be an extremely challenging task. The identification of the minor obstruction sets for the parameter of tree-depth and the computation of the immersion obstruction sets for graph classes closed under taking of immersions are the subjects that we deal with in the next two chapters.

For more on algorithmic techniques derived from the Graph Minor

Theory see Chapter 8.

CHAPTER 4

Identifying the Obstructions for Tree-depth

As we saw in the previous chapter, for every graph class $\mathcal{C}$ that is closed under taking of minors or immersions there exists a cubic time algorithm deciding the membership of a graph G in the graph class $\mathcal{C}$. However, recall that the construction of such an algorithm requires an explicit description of the set $\mathbf{obs}_{\leq_m}(\mathcal{C})$ or $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ respectively.

Our goal in this chapter is to study ways of identifying minor obstruction sets. In particular, in this chapter, we would like to show how challenging the identification of an obstruction set can be by trying to identify the minor obstructions for the parameter of tree-depth.

The graph parameter of tree-depth (also known as the vertex ranking problem [21], or the ordered coloring problem [132]) has received much attention, mostly because of the theory of graph classes of bounded expansion, developed by Nešetřil and Ossona de Mendez. See, for example, [160–169]. (For extensive details on the graph classes of bounded expansion, see [170].) Furthermore, the tree-depth of a graph is equivalent to the minimum-height of an elimination tree of a graph [45, 57, 166] (this measure is of importance for the parallel Cholesky factorization of matrices [147]). Let $\mathcal{G}_k$ be the class of all graphs with tree-depth at most k.

In the first part of this section, we examine the sets $\mathbf{obs}_{\leq_{\mathrm{m}}}(\mathcal{G}_k)$, $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$, and $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$, where by $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ and $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ we denote the minimal graphs according to the subgraph and induced subgraph ordering respectively that do not belong to $\mathcal{G}_k$. From Theorem 3.5, it follows that $\mathbf{obs}_{\leq_{\mathrm{m}}}(\mathcal{G}_k)$ is finite for each $k \geq 0$. The finiteness of $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ follows from results in [166]. Also it is easy to verify that $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ is finite (see Observation 4.3).

Our first result is an upper bound of $2^{2^{k-1}}$ to the order of the graphs in $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ for $k \geq 0$. This bound also holds for $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ and $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$ as $\mathbf{obs}_{\leq_m}(\mathcal{G}_k) \subseteq \mathbf{obs}_{\subseteq}(\mathcal{G}_k) \subseteq \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ (Observation 4.2). Our next result is a structural lemma that constructs new obstructions from simpler ones. This permits us to identify, for each $k \geq 0$, all acyclic obstructions of the graph class $\mathcal{G}_k$. Evenmore, by using this characterization we prove that the acyclic obstructions are exactly $\frac{1}{2}2^{2^{k-1}-k}(1+2^{2^{k-1}-k})$ for all relations. So far, such a parameterized set of acyclic obstructions is known only for classes of bounded pathwidth [219] and variations of it such as search number [179], proper-pathwidth [219], linear-width [220] (see [213] for similar results on graphs with bounded feedback vertex set number). For general results on obstruction sets see also [3, 35, 61, 143, 181]. However, this is the first time that an exact enumeration of parameterized obstructions has been derived. Our next result is the identification of the sets $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$, $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$, and $\mathbf{obs}_{\sqsubseteq}(G_k)$, for $k \leq 3$. For $k = 3$, these sets have 12, 14, and 29 graphs, respectively. Finally, we show a theorem which allows us to identify proper subgraphs of a graph G, which have the same tree-depth as G.

For a study of the parameter of tree-depth from the scope of Graph Searching, see Chapter 10.

4.1 An Introduction to Tree-depth

The tree-depth of a connected graph G, denoted by $\mathbf{td}(G)$, is defined as follows.

Definition 4.1 (Tree-depth)**.**

$$\mathbf{td}(G) = \begin{cases} 1 & \text{if } |V(G)| = 1 \\ 1 + \min_{v \in V(G)} \mathbf{td}(G \setminus v) & \text{if } |V(G)| > 1 \end{cases}$$

In the case where G is not connected, then $\mathbf{td}(G) = \max\{\mathbf{td}(H) \mid H \in \mathcal{C}(G)\}$. (Recall that by $\mathcal{C}(G)$ we denote the set of the connected components of G.)

We also say that a graph G admits a *k-vertex ranking* if there exists a proper coloring $\rho : V(G) \to \{1, \ldots, k\}$ such that every (x, y)-path between two vertices where $\rho(x) = \rho(y)$ contains a vertex z with $\rho(z) > \rho(x)$.

It was proven by J. Nešetřil and P. Ossona de Mendez that.

Lemma 4.1 ([166])**.** *The tree-depth of a graph G is equal to the minimum integer k such that G admits a k-vertex ranking.*

Using induction, one can prove that.

Lemma 4.2 ([166])**.** *For any non-negative integer n,*

$$\mathbf{td}(P_n) = \lceil \log_2(n + 1) \rceil.$$

For every non-negative integer k, we denote by $\mathcal{G}_k$ the class of graphs with tree-depth at most k, that is, $\mathcal{G}_k = \{G \mid \mathbf{td}(G) \leq k\}$.

It is known that.

Lemma 4.3 ([21, 166]). *If H is a minor of G, then $\mathbf{td}(H) \leq \mathbf{td}(G)$.*

Thus, Lemma 4.3 has as a direct consequence that for any non-negative integer k, $\mathcal{G}_k$ is minor-closed. Therefore, by Theorem 3.5, the set $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$ is finite.

For every $\mathsf{R} \in \{\sqsubseteq, \subseteq\}$, we denote by $\mathbf{obs}_{\mathsf{R}}(\mathcal{G}_k)$ the set of the graphs with tree-depth strictly bigger than k that are minimal with respect to the relation R.

Before stating the next lemma we need to give the following definitions. Given two graphs G and H, we say that H is *homomorphic* to G if there is a mapping $f : V(H) \to V(G)$ (called *homomorphism*) such that for every $v, u \in V(H)$, if $\{u, v\} \in E(H)$ then $\{f(u), f(v)\} \in E(G)$. Moreover, we say that G and H are *isomorphic* if there is a bijection $g : V(H) \to V(G)$ (called *isomorphism*) such that for every $v, u \in V(H)$, $\{u, v\} \in E(H)$ if and only if $\{f(u), f(v)\} \in E(G)$. In the special case where G and H are the same graph the isomorphism is also called *automorphism*. We say that two graphs G_1, G_2 are *hom-equivalent* if G_1 is homomorphic to G_2 and G_2 is homomorphic to G_1. Even, an automorphism f of a graph is said to be *involutive* if and only if $f \circ f = \mathbf{id}$. Finally, given a graph G and a function $\lambda : V(G) \to \mathbb{N}$ we say that an automorphism $f : V(G) \to V(G)$ is λ-preserving if $f \circ \lambda = \lambda$ and we say that $f : V(G) \to V(G)$ has the *fixed point property* if, for any connected subgraph H of G, $f(H) \cap H = \emptyset$ or contains a fixed point of f.

Lemma 4.4 ([166]). *There exists a function $F : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ with the following property: For any integer N, any graph G of order $n >$*

$F(N, \mathbf{td}(G))$, and any mapping $g : V(G) \to [N]$, there exists a non-trivial involutive g-preserving automorphism $\mu : G \to G$ with the fixed point property.

An immediate corollary of the above lemma is the following.

Lemma 4.5 ([166])**.** *Let $k \geq 1$ be an integer. Then, the graph class $\mathcal{G}_k$ includes a finite subset $\hat{\mathcal{G}}_k$ such that, for every graph $G \in \mathcal{G}_k$, there exists $\hat{G} \in \hat{\mathcal{G}}_k$ which is hom-equivalent to G and isomorphic to an induced subgraph of G.*

Furthermore, from Lemma 4.4, a tower function bound can be derived for the order of the forbidden subgraphs.

As the purpose of this chapter is to study ways of computing obstruction sets we would like to mention here that when we are given an upper bound on the order of the obstructions of a graph class $\mathcal{F}$ and an algorithm deciding whether an input graph belongs to $\mathcal{F}$ then we may compute the obstruction set of $\mathcal{F}$. Indeed, notice that this can be done by enumerating all the graphs up to this bound and deciding whether they belong to $\mathcal{F}$ or not. Then, the minimal graphs not belonging to $\mathcal{F}$ are the obstructions of $\mathcal{F}$. Thus, for the computation of the minor obstruction set of $\mathcal{G}_k$, all we need is an algorithm deciding whether a graph G has tree-depth k.

Theorem 4.1 ([21])**.** *For any fixed integer k, there is a polynomial time algorithm deciding whether a graph G has tree-depth k.*

However, as stated before, the bound obtained from Lemma 4.4 is a tower function of k. We prove, in the next section, that a direct argument shows a much better bound. Moreover, in the next sections, by combinatorial arguments we identify the acyclic graphs in $\mathbf{obs}_{\mathsf{R}}(\mathcal{G}_k)$, $k \geq 1$ and $\mathsf{R} \in \{\leq_m, \subseteq, \sqsubseteq\}$, and the sets $\mathbf{obs}_{\mathsf{R}}(\mathcal{G}_k)$ for $k \in [3]$ and $\mathsf{R} \in \{\leq_m, \subseteq, \sqsubseteq\}$.

4.2 Upper Bound on the Order of the Obstructions for $\mathcal{G}_k$

Observation 4.1. *For every $k \geq 0$, all graphs in $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k), \mathbf{obs}_{\subseteq}(\mathcal{G}_k)$, and $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$ are connected.*

Proof. Follows directly from the fact that for any graph G, $\mathbf{td}(G) = \max\{\mathbf{td}(C) \mid C \in \mathcal{C}(G)\}$. □

Observation 4.2. *For every non-negative integer k,*

$$\mathbf{obs}_{\leq_m}(\mathcal{G}_k) \subseteq \mathbf{obs}_{\subseteq}(\mathcal{G}_k) \subseteq \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k).$$

Observation 4.3. *Let G be a graph such that $G \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$, for some integer k. Then there exists $G' \in \mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ such that $V(G) = V(G')$ and $E(G') \subseteq E(G)$.*

Proof. Let G be a counterexample of minimal size. Then there exists an edge e such that $G' = G \setminus e$ also belongs to $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ and $V(G) = V(G')$ and $E(G') \subseteq E(G)$. □

By Lemma 4.5 and Observation 4.3 it is easy to see that, for every positive integer k, the sets $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ and $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ are also finite.

Theorem 4.2. *For any integer $k > 0$, if G is a graph with $\mathbf{td}(G) > k$, then G contains a connected subgraph H with $\mathbf{td}(H) > k$ and $|V(H)| \leq 2^{2^{k-1}}$.*

Proof. We may assume that G is connected, otherwise from the definition we focus on the component of G that determines its tree-depth. Also, without loss of generality, let $\mathbf{td}(G) = k + 1$. We prove the statement by induction:

If $\mathbf{td}(G) = 2$, then G contains at least one edge, and we may set $H = K_2$. If $\mathbf{td}(G) = 3$, then G is not a star forest, that is, it contains either P_4 or K_3 as a subgraph.

Suppose now that $\mathbf{td}(G) = k + 1$, for $k \geq 3$, and assume that the statement holds for all smaller values of tree-depth. If G contains P_{2^k} as a subgraph, then we may set $H = P_{2^k}$. Otherwise, each two vertices in G are connected by a path of length at most $2^k - 2$.

Since $\mathbf{td}(G) > k - 1$, by induction hypothesis, G contains a subgraph H_0 with $\mathbf{td}(H_0) \geq k$ and $m \leq 2^{2^{k-2}}$ vertices $v_1, \ldots, v_m$. For each $i = 1, \ldots, m$, the graph $G \setminus v_i$ has tree-depth greater than $k - 1$. Hence $G \setminus v_i$ contains a subgraph H_i with at most $2^{2^{k-2}}$ vertices and tree-depth at least k.

If there exists i such that $V(H_0) \cap V(H_i) = \emptyset$, then we let H consist of H_0, H_i and the shortest path that connects them. For every vertex v of H, the graph $H \setminus v$ contains H_0 or H_i as a subgraph, hence the tree-depth of $H \setminus v$ is at least k and $\mathbf{td}(H) > k$. Also, $|V(H)| \leq 2^{2^{k-2}+1} + 2^k - 3 \leq 2^{2^{k-1}}$ (for $k \geq 3$).

On the other hand, if all the graphs H_i intersect H_0, then we set $H = H_0 \cup H_1 \cup \ldots \cup H_m$. Since all the graphs H_i are connected, the graph H is connected as well, and it has at most $m + m(2^{2^{k-2}} - 1) \leq 2^{2^{k-1}}$ vertices. Similarly to the previous case, the graphs $H \setminus v_i$ contain H_i as a subgraph (for $i = 1, \ldots, m$), and the graph $H \setminus v$ for v different from v_1, $\ldots$, v_m contains H_0 as a subgraph, hence $\mathbf{td}(H) > k$. $\square$

From Theorem 4.2 and Observations 4.2 and 4.3 we obtain the following corollary,

Corollary 4.1. *All graphs in* $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ *(and therefore, also in* $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ *and* $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$*) have at most* $2^{2^{k-1}}$ *vertices.*

4.3 A Structural Lemma for the Obstructions of Tree-depth

In this section, we prove a lemma for tree-depth that permits us to build obstructions from simpler ones. We, first, consider the following observations.

Observation 4.4. *Let G be a connected graph such that $\mathbf{td}(G) = k$ and $\rho : V(G) \to [k]$ a k-vertex ranking of G. Then $|\rho^{-1}(k)| = 1$.*

Proof. If v_1 and v_2 are two (non-adjacent) vertices in $\rho^{-1}(k)$, then there exists a path with endpoints v_1, v_2. Observe that all internal vertices of this path have color smaller than k, a contradiction. □

Observation 4.5. *If $G \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ (or $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ or $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$) then, for every $v \in V(G)$, there exists a $(k+1)$-vertex ranking ρ such that $\rho(v) = k+1$.*

Proof. As $G \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ (or $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ or $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$), $G \setminus v$ admits a k-vertex ranking ρ. Then $\rho \cup (v, k+1)$ is the required $(k+1)$-vertex ranking of G. □

Let G_1 and G_2 be two disjoint graphs and let $v_i \in V(G_i)$, for $i \in [2]$. We define

$$\mathbf{j}(G_1, G_2, v_1, v_2) = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{\{v_1, v_2\}\}).$$

Observation 4.6. *Let G_1 and G_2 be disjoint graphs where $\mathbf{td}(G_1) \leq k$ and $\mathbf{td}(G_2) \leq k$. Let $v_i \in V(G_i)$, $i \in [2]$. Then the graph $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ has tree-depth at most $k+1$.*

Proof. Let ρ_i be a k-vertex ranking of G_i, $i \in [2]$. Then $\rho = \rho_1 \cup \rho_2 \setminus \{(v_1, \rho_1(v_1))\} \cup \{(v_1, k+1)\}$ is a $(k+1)$-vertex ranking of G. □

Observation 4.7. *Let G_1 and G_2 be disjoint connected graphs such that $\mathbf{td}(G_1) \geq k$ and $\mathbf{td}(G_2) \geq k$. Let $v_i \in V(G_i)$, $i \in [2]$. Then, the graph $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ has tree-depth at least $k+1$.*

Proof. Assume, in contrary, that there exists a k-vertex ranking $\rho : V(G) \to [k]$. Notice that $\rho^{-1}(k) \neq \emptyset$, otherwise $\mathbf{td}(G) < k$ contradicting the fact that $\mathbf{td}(G_1) \geq k$. Combining this fact with Observation 4.4, G has a unique vertex v where $\rho(v) = k$. Without loss of generality, we assume that $v \in V(G_1)$. Then, the restriction of ρ to G_2 gives a $(k-1)$-vertex ranking of it, a contradiction. □

Lemma 4.6. *Let k be a positive integer and let $\mathsf{R} \in \{\sqsubseteq, \subseteq, \leq_m\}$. Let G_1 and G_2 be disjoint graphs such that $G_1, G_2 \in \mathbf{obs}_{\mathsf{R}}(\mathcal{G}_{k-1})$ and let $v_1 \in V(G_1), v_2 \in V(G_2)$. Then $\mathbf{j}(G_1, G_2, v_1, v_2) \in \mathbf{obs}_{\mathsf{R}}(\mathcal{G}_k)$.*

Proof. Let G_1 and G_2 such that $G_1, G_2 \in \mathbf{obs}_{\mathsf{R}}(\mathcal{G}_{k-1})$ and let $v_i \in V(G_i)$, $i \in [2]$. We set $G = \mathbf{j}(G_1, G_2, v_1, v_2)$. We first prove that $\mathbf{td}(G) = k+1$. Indeed, Observation 4.6 yields $\mathbf{td}(G) \leq k+1$ and Observation 4.7 yields $\mathbf{td}(G) \geq k+1$.

We now have to prove that if G' is the result of the removal or the contraction of some edge e in G, then $\mathbf{td}(G') \leq k$ (this also covers the case of a vertex removal as, from the way G was defined, G is connected and thus the removal of a vertex implies the removal of at least one edge).

We examine first the case where $e = \{v_1, v_2\}$. If $G' = G \setminus e$, then from the definition, $\mathbf{td}(G) = \max\{\mathbf{td}(G_1), \mathbf{td}(G_2)\} \leq k$. If $G' = G \mathbin{/} e$, then from Observation 4.5, there exists a k-vertex ranking ρ_i of G_i such that $\rho_i(v_i) = k$, $i \in [2]$. Then if v_{new} is the result of the contraction of e we have that $\rho : V(G') \to [k]$ where

$$\rho(x) = \begin{cases} \rho_1(x) & \text{if } x \in V(G_1) \setminus \{v_1\} \\ \rho_2(x) & \text{if } x \in V(G_2) \setminus \{v_2\} \\ k & \text{if } x = v_{\text{new}} \end{cases}$$

is a k-vertex ranking of G', therefore $\mathbf{td}(G') \leq k$.

Finally, we examine the case where e is an edge of G_1 or G_2. Without loss of generality we assume that $e_1 \in E(G_1)$. Because $G_1 \in \mathbf{obs}_{\subseteq}(\mathcal{G}_{k-1})$, there exists a $(k-1)$-vertex ranking ρ'_1 of $G_1 \setminus e$ (and $G_1 \ / \ e$). From Observation 4.5, since $G_2 \in \mathbf{obs}_{\subseteq}(\mathcal{G}_{k-1})$, there exists a k-vertex ranking ρ_2 of G_2 such that $\rho_2(v_2) = k$. It is easy to see that $\rho'_1 \cup \rho_2$ is a k-vertex ranking of G', thus $\mathbf{td}(G') \leq k$ and this completes the proof of the lemma. □

4.4 Acyclic obstructions for tree-depth

For every integer $k \geq 0$, we recursively define the graph class $\mathcal{T}_k$ as follows. Let $\mathcal{T}_0 = \{K_1\}$ and, for every $k \geq 1$, we set

$$\mathcal{T}_k = \{\mathbf{j}(G_1, G_2, v_1, v_2) \mid G_1, G_2 \in \mathcal{T}_{k-1}, v_i \in V(G_i), i \in [2]\}$$

The above definition permits us to state Lemma 4.6 as follows.

Observation 4.8. *For every positive integer k and every $\mathsf{R} \in \{\sqsubseteq, \subseteq, \leq_m\}$, $\mathcal{T}_k \subseteq \mathbf{obs}_{\mathsf{R}}(\mathcal{G}_k)$.*

Lemma 4.7. *For any positive integer k, if $G \in \mathcal{T}_k$, then for any vertex $v \in V(G)$ there exists a leaf $u \neq v$ of G such that the tree created from $G \setminus u$ by adding a leaf adjacent to v also belongs to $\mathcal{T}_k$.*

Proof. Assume, that this holds for any tree in $\mathcal{T}_{k-1}$, $k \geq 2$. Let $G_1, G_2 \in \mathcal{T}_{k-1}$ and $v_i \in V(G_i)$, $i \in [2]$, such that $G = \mathbf{j}(G_1, G_2, v_1, v_2)$. Consider an arbitrary vertex $v \in V(G)$, and let us show that there exists a leaf u of G that we can move to v while preserving membership in $\mathcal{T}_k$. Without loss of generality, we may assume that $v \in V(G_1)$. By the induction hypothesis, there exists a vertex $u' \in V(G_1)$ such that the tree created from $G_1 \setminus u'$ by adding a leaf adjacent to v is also in $\mathcal{T}_{k-1}$. If $u' \neq v_1$ we may set $u = u'$. Otherwise, let u'' be the leaf of G_2 that can be moved to v_2. In this case, we can set $u = u''$: Moving the leaf u'' to v has the same result as moving it to v_2, moving the leaf u' to v, and replacing the edge e by an edge between u'' and the vertex of G_1 that used to be adjacent to u'. $\square$

In Lemma 4.6 we described a procedure that for any non-negative integer k constructs graphs $G \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k+1})$ from disjoint graphs $G_1, G_2 \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ (by adding an edge that connects a vertex v_1 of G_1 and a vertex v_2 of G_2). With the following lemma we fully characterize and construct all the acyclic graphs in $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k+1})$ for every non-negative integer k.

Lemma 4.8. *Let G be a tree in $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ for $k \geq 1$. Then there exists an edge $e \in E(G)$ such that if $\{G_1, G_2\} = \mathcal{C}(G \setminus \{e\})$ then $G_1, G_2 \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k-1})$.*

Proof. We examine the non-trivial case where $k \geq 2$ assuming that the statement holds for all acyclic obstructions of smaller tree-depth. From Observation 4.6, we obtain that for each edge $e = \{v_1, v_2\} \in E(G)$, at least one of the connected components G_1, G_2 of $G \setminus e$ has tree-depth at least k. We claim that G contains at least one edge $e = \{v_1, v_2\}$ such that both connected components of $G \setminus e$ have tree-depth k. Suppose that this is not correct. Then we can direct each edge $e = \{v_1, v_2\}$ of $E(G)$ such that its tail belongs to the connected component of $G \setminus e$ that has tree-depth strictly less than k. We denote this directed tree by $\tilde{T}$. As $k \geq 2$, $\tilde{T}$ contains internal vertices. Moreover, all edges of $\tilde{T}$ that are incident to a leaf are directed away from it. It follows that $\tilde{T}$ contains an internal vertex v of out-degree 0. This means that each, say G_i, connected component of $G \setminus v$ has a $(k-1)$-vertex ranking ρ_i. Then $\rho = \{(v,k)\} \cup \bigcup_{i=1,\ldots,m} \rho_i$ is a k-vertex ranking of G, a contradiction and this completes the proof of the claim.

Let now G_i be the connected component of $G \setminus e$ that contains v_i, $i \in [2]$. If one, say G_1, is not in $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k-1})$ then it contains an induced subgraph G_1' such that $G_1' \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k-1})$. Additionally, there is a unique path P in G that connects G_1' with G_2. Observe that since $G \in \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k-1})$, G is exactly the union of G_1', G_2 and P. We need to show that P has no inner vertices. Suppose that this is not the case, and let w be the inner vertex of P adjacent to a vertex $v \in V(G_1)$. By the induction hypothesis, G_1' and G_2 satisfy the conditions of Lemma 4.7, thus G_1 contains a leaf u such that the graph obtained from G_1 by moving t he leaf u to v belongs to $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_{k-1})$. This implies that we may remove the vertex u from G and consider w to be its replacement. The created graph is a proper induced subgraph of G and has tree-depth $k+1$, a contradiction. This completes the proof of the lemma. □

Observe now that the following is a direct consequence of Lemmata 4.6 and 4.8.

Theorem 4.3. *Let k be a non-negative integer. Then $\mathcal{T}_k$ is the set of all acyclic graphs in $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$.*

Corollary 4.2. *For every non-negative integer k, $\mathcal{T}_k$ is the set of all acyclic graphs in $\mathbf{obs}_{\subseteq}(\mathcal{G}_k)$ (or in $\mathbf{obs}_{\leq_m}(\mathcal{G}_k)$). (See Figure 4.1)*

Proof. Follows directly from Observations 4.2 and 4.8. □

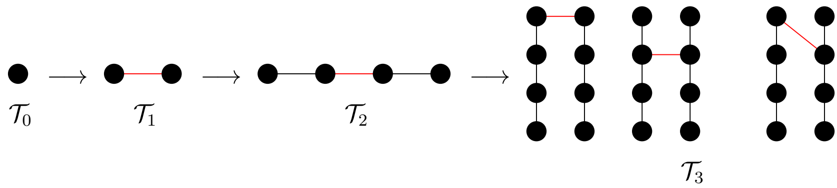


Figure 4.1: Examples of acyclic obstructions.

4.5 Lower Bound on the Number of the Obstructions for $\mathcal{G}_k$

In this section, we prove that $|\mathcal{T}_k| = \frac{1}{2}2^{2^{k-1}-k}(1 + 2^{2^{k-1}-k}), k \geq 1$. This gives a lower bound on $|\mathbf{obs}_{\leq_m}(\mathcal{G}_k)|, k \geq 2$. As we shall see later we can identify the elements of the sets $\mathbf{obs}_{\subseteq}(\mathcal{G}_i)$, $\mathbf{obs}_{\leq_m}(\mathcal{G}_i)$, and $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_i)$, for $i = 0, 1, 2, 3$.

For a tree $G \in \mathcal{T}_k$ such that $G = \mathbf{j}(G_1, G_2, v_1, v_2)$, we call v_1v_2 the *middle edge* of G.

Observation 4.9. *If k is a non-negative integer then every graph in $\mathcal{T}_k$ has exactly 2^k vertices. This implies that the middle edge of a graph $G \in \mathcal{T}_k$ is unique.*

Consider also the following.

Observation 4.10. *Let T^1, T^2 be two trees and $e^i = \{v_1^i, v_2^i\} \in E(T^i)$, $i \in [2]$. If ϕ is an isomorphism from T^1 to T^2 such that $\phi(v_i^1) = v_i^2$, $i \in [2]$, and T_i^j is the connected component of $T^j \setminus e^j$ that contains v_i^j, $i \in [2]$, $j \in [2]$, then $\phi_i = \{(x, y) \in \phi \mid x \in V(T_i^1)\}$ is an isomorphism from T_i^1 to T_i^2, $i \in [2]$.*

It is easy to see that the automorphisms of a graph form a group. We use notation $\mathbf{Aut}(G)$ for the automorphism group of a graph G. Observation 4.10 easily implies the following.

Observation 4.11. *Let T be a tree and $e = \{v_1, v_2\} \in E(T)$. If $\phi \in \mathbf{Aut}(T)$ satisfies $\phi(v_i) = v_{3-i}$, $i \in [2]$, and T_i is the connected component of $T \setminus e$ that contains v_i, $i \in [2]$, then $\phi' = \{(x, y) \in \phi \mid x \in V(T_1)\}$ is an isomorphism from T_1 to T_2.*

Observation 4.12. *Let G_1, G_2 be disjoint graphs such that $G_1, G_2 \in \mathcal{T}_k$, $k \geq 1$, and $v_i \in V(G_i)$, $i \in [2]$. If $\phi \in \mathbf{Aut}(G)$, where $G = \mathbf{j}(G_1, G_2, v_1, v_2)$, then $\phi(e) = e$.*

Proof. Follows directly from Observation 4.9. □

Lemma 4.9. *Let $G \in \mathcal{T}_k$ for $k \geq 1$, $e = \{v_1, v_2\} \in E(G)$ the middle edge and $\phi \in \mathbf{Aut}(G)$. If there exists $v \in V(G)$ such that $\phi(v) = v$, then $\phi(v_i) = v_i$, $i \in [2]$.*

Proof. We examine the non-trivial case where $k \geq 2$. Suppose, in contrary, that $\phi(v_i) = v_{3-i}$, $i \in [2]$. We denote by G_1, G_2 the connected components of $G \setminus e$ where, without loss of generality, $v, v_1 \in V(G_1)$. From Observation 4.11, $\phi' = \{(v_1, v_2) \in \phi \mid v_1 \in V(G_1)\}$ is an isomorphism of G_1 to G_2, a contradiction since $\phi'(v) = \phi(v) = v$. □

We now proceed to the proof of the following.

Lemma 4.10. *Let k be a non-negative integer. For any $G \in \mathcal{T}_k$ and $\phi \in \mathbf{Aut}(G)$, if there exists $v \in V(G)$ such that $\phi(v) = v$ then $\phi = \mathbf{id}$.*

Proof. We use induction on k. For $k = 0$ the claim is trivial. Assume now that the claim holds for $k = n \geq 0$. Let $k = n + 1$. We denote by $e =$

$\{v_1, v_2\} \in E(G)$ the middle edge and by G_1, G_2 the connected components of $G \setminus e$, where $v_i \in V(G_i)$, $i \in [2]$. Since $\phi \in \mathbf{Aut}(G)$, from Lemma 4.9, it follows that $\phi(v_i) = v_i$, $i \in [2]$. Hence ϕ is an isomorphism from $G \setminus e$ to $G \setminus e$. From Observation 4.10, $\phi_i = \{(v, u) \in \phi \mid v \in V(G_i)\} \in \mathbf{Aut}(G_i)$, $i \in [2]$. Observe that $\phi_i(v_i) = \phi(v_i) = v_i$, $i \in [2]$. Since $G_i \in \mathcal{T}_n$, $i \in [2]$, by the induction hypothesis, ϕ_i, $i \in [2]$, is the trivial automorphism of G_i. Therefore, $\phi = \mathbf{id}$. $\square$

Let G be a graph and $v \in V(G)$. We denote by $\mathbf{tr}_G(v)$ the *orbit* of the automorphism group of G that contains v, that is,

$$\mathbf{tr}_G(v) = \{u \in V(G) \mid \exists \phi \in \mathbf{Aut}(G) \text{ such that } \phi(u) = v\}.$$

Lemma 4.11. *Let G_1, G_2 be disjoint graphs such that $G_1, G_2 \in \mathcal{T}_k$, $v_2, v_2' \in V(G_2)$ such that $v_2 \in \mathbf{tr}_{G_2}(v_2')$ and $v_1 \in V(G_1)$. Then $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ and $G' = \mathbf{j}(G_1, G_2, v_1, v_2')$ are isomorphic.*

Proof. Let $\mathbf{id} \in \mathbf{Aut}(G_1)$ and $\phi \in \mathbf{Aut}(G_2)$, such that $\phi(v_2) = v_2'$. Then $\mathbf{id} \cup \phi$ is an isomorphism from G to G'. $\square$

Lemma 4.12. *Let G_1, G_2 be disjoint graphs such that $G_1, G_2 \in \mathcal{T}_k$, $v_2, v_2' \in V(G_2)$ such that $v_2 \notin \mathbf{tr}_{G_2}(v_2')$ and $v_1 \in V(G_1)$. Then $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ and $G' = \mathbf{j}(G_1, G_2, v_1, v_2')$ are not isomorphic.*

Proof. Assume, in contrary, that ϕ is an isomorphism from G to G'. Observation 4.12 implies that either $\phi(v_1) = v_1$ and $\phi(v_2) = v_2'$ or $\phi(v_1) = v_2'$ and $\phi(v_2) = v_1$. We first exclude the case where $\phi(v_1) = v_1$ and $\phi(v_2) = v_2'$. Indeed, from Observation 4.10, $\phi' = \{(x, y) \in \phi \mid x \in V(G_2)\} \in \mathbf{Aut}(G_2)$ and moreover $\phi'(v_2) = \phi(v_2) = v_2'$, a contradiction since $v_2 \notin \mathbf{tr}_{G_2}(v_2')$. Thererefore, $\phi(v_1) = v_2'$ and $\phi(v_2) = v_1$. From Observation 4.10,

$\phi_i = \{(x,y) \in \phi \mid x \in V(G_i)\}$ is an isomorphism from G_i to G_{3-i}, $i \in [2]$. Then $\psi = \phi_1 \circ \phi_2 \in \mathbf{Aut}(G_2)$ and $\psi(v_2) = \phi_1(\phi_2(v_2)) = \phi_1(\phi(v_2)) = \phi_1(v_1) = v_2'$. It follows that $v_2 \in \mathbf{tr}_{G_2}(v_2')$, a contradiction. $\square$

Given a graph G, we say that G is *asymmetric* if it has a trivial automorphism group. Moreover, we say that a graph G is *2-asymmetric* if its only non-trivial automorphism is an involution without fixed points.

Lemma 4.13. *Let k be a non-negative integer and let G_1, G_2 be two disjoint non-isomorphic graphs such that $G_1, G_2 \in \mathcal{T}_k$. Then the graph $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ is asymmetric.*

Proof. Suppose that $\phi \in \mathbf{Aut}(G)$ and $\phi \neq \mathbf{id}$. From Lemma 4.10, $\phi(v) \neq v$ for all $v \in V(G)$ and from Observation 4.12, $\phi(v_i) = v_{3-i}$, $i \in [2]$. From Observation 4.11, G_1 is isomorphic to G_2, a contradiction. $\square$

Lemma 4.14. *Let k be a non-negative integer and let G_1, G_2 be two disjoint graphs such that $G_1, G_2 \in \mathcal{T}_k$. If ϕ is an isomorphism from G_1 to G_2 and $v_i \in V(G_i)$, $i \in [2]$, such that $\phi(v_1) \notin \mathbf{tr}_{G_2}(v_2)$, then $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ is asymmetric.*

Proof. Suppose that $\psi \in \mathbf{Aut}(G)$ and $\psi \neq \mathbf{id}$. From Lemma 4.10, $\psi(v) \neq v$ for all $v \in V(G)$ and from Observation 4.12, $\psi(v_1) = v_2$ and $\psi(v_2) = v_1$. From Observation 4.11, $\chi = \{(x,y) \in \psi \mid x \in V(G_1)\}$ is an isomorphism from G_1 to G_2. Moreover, $\phi \circ \chi^{-1}$ is an automorphism of G_2 mapping v_2 to $\phi(v_1)$, contradicting the assumption that $\phi(v_1) \notin \mathbf{tr}_{G_2}(v_2)$. $\square$

Lemma 4.15. *Let k be a non-negative integer and let G_1, G_2 be two disjoint graphs such that $G_1, G_2 \in \mathcal{T}_k$. If $\phi : V(G_1) \to V(G_2)$ is an isomorphism from G_1 to G_2 and $v_i \in V(G_i), i \in [2]$, are two vertices such that $\phi(v_1) \in \mathbf{tr}_{G_2}(v_2)$, then $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ is 2-asymmetric.*

Proof. Since $\phi(v_1) \in \mathbf{tr}_{G_2}(v_2)$, there exists an isomorphism $\psi : V(G_1) \to V(G_2)$ such that $\psi(v_1) = v_2$. Observe that $\chi = \psi \cup \psi^{-1}$ is an automorphism of G, and that χ is an involution without fixed points. Consider an automorphism $\chi' \neq \mathbf{id}$ of G. From Lemma 4.10 and Observation 4.12, $\chi'(v_1) = v_2$ and from Observation 4.11, $\chi'_1 = \{(x, y) \in \chi' | x \in V(G_1)\}$ is an isomorphism of G_1 and G_2. Then $\chi'_1 \circ \psi^{-1}$ is an automorphism of G_2 that fixes v_2, and from Lemma 4.10, $\chi'_1 = \psi$. We conclude that $\chi' = \chi$, and thus $\mathbf{Aut}(G) = \{\mathbf{id}, \chi\}$ and G is 2-asymmetric. □

From Theorem 4.3 and Lemmata 4.13, 4.14, and 4.15 follows directly that.

Observation 4.13. *If G is a graph such that $G \in \mathcal{T}_k$ then G is either asymmetric or 2-asymmetric.*

For every integer $k \geq 0$, we define the following partition of $\mathcal{T}_k$:

$$\mathcal{A}_k = \{G \in \mathcal{T}_k \mid \mathbf{Aut}(G) = \{\mathbf{id}\}\} \text{ and } \mathcal{B}_k = \{G \in \mathcal{T}_k \mid \mathbf{Aut}(G) \neq \{\mathbf{id}\}\}.$$

We denote $\alpha_k = |\mathcal{A}_k|$, $\beta_k = |\mathcal{B}_k|$, and $\tau_k = |\mathcal{T}_k| = \alpha_k + \beta_k$. We also set $\gamma_k = 2^{k-2}$. A direct consequence of Observations 4.9 and 4.13 is the following.

Observation 4.14. *Let $k \geq 2$ be an integer. Then the automorphism group of each graph $G \in \mathcal{A}_k$ (respectively $G \in \mathcal{B}_k$) has exactly γ_{k+2} (respectively γ_{k+1}) orbits.*

Observation 4.15. *$\beta_0 = \alpha_1 = \alpha_2 = 0$ and $\alpha_0 = \beta_1 = \beta_2 = 1$.*

Theorem 4.4. *For every integer $k \geq 1$, $\tau_k = 2^{2^k-(2k+1)} + 2^{2^{k-1}-(k+1)}$.*

Proof. First observe that for $k \in [2]$ the claim holds. Let G be a graph. Recall that $G \in \mathcal{T}_k$ if and only if $G = \mathbf{j}(G_1, G_2, v_1, v_2)$ for some $G_i \in \mathcal{T}_{k-1}$, and $v_i \in V(G_i)$, $i \in [2]$. Therefore, in order to count τ_k it is sufficient to count the ways to choose $G_1, G_2 \in \mathcal{T}_{k-1}$ and $v_i \in V(G_i)$, $i \in [2]$, and not end up with isomorphic graphs. Let G_1, G_2 be graphs such that $G_i \in \mathcal{T}_{k-1}$ and $v_i \in V(G_i)$, $i \in [2]$. We define

$$\begin{aligned}
\mathcal{A}_k^1 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \not\simeq G_2, G_i \in \mathcal{A}_{k-1} \\
&\quad \text{and } v_i \in V(G_i), i \in [2]\} && (4.1)\\
\mathcal{A}_k^2 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \not\simeq G_2, G_i \in \mathcal{B}_{k-1} \\
&\quad \text{and } v_i \in V(G_i), i \in [2]\} && (4.2)\\
\mathcal{A}_k^3 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \not\simeq G_2, G_1 \in \mathcal{A}_{k-1}, G_2 \in \mathcal{B}_{k-1}, \\
&\quad \text{and } v_i \in V(G_i), i \in [2]\} && (4.3)\\
\mathcal{A}_k^4 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \simeq_\phi G_2, G_i \in \mathcal{A}_{k-1}, \\
&\quad \text{and } v_i \in V(G_i), i \in [2], \text{ such that } \phi(v_1) \notin \mathbf{tr}_{G_2}(v_2)\} && (4.4)\\
\mathcal{A}_k^5 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \simeq_\phi G_2, G_i \in \mathcal{B}_{k-1}, \\
&\quad \text{and } v_i \in V(G_i), i \in [2], \text{ such that } \phi(v_1) \notin \mathbf{tr}_{G_2}(v_2)\} && (4.5)\\
\mathcal{B}_k^1 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \simeq_\phi G_2, G_i \in \mathcal{A}_{k-1}, \\
&\quad \text{and } v_i \in V(G_i), i \in [2], \text{ such that } \phi(v_1) \in \mathbf{tr}_{G_2}(v_2)\} && (4.6)\\
\mathcal{B}_k^2 &= \{G \mid G = \mathbf{j}(G_1, G_2, v_1, v_2), G_1 \simeq_\phi G_2, G_i \in \mathcal{B}_{k-1}, \\
&\quad \text{and } v_i \in V(G_i), i \in [2], \text{ such that } \phi(v_1) \in \mathbf{tr}_{G_2}(v_2)\}. && (4.7)
\end{aligned}$$

By their definitions, the above sets are a partition of $\mathcal{T}_k$. From Lemma 4.13 (for Relations (4.1)–(4.3)) and from Lemma 4.14 (for Relations (4.4) and (4.5)), the union of the first five is a subset of $\mathcal{A}_k$. Moreover, from Lemma 4.15 (applied to Relations (4.6) and (4.7)) the union of the last two is a subset of $\mathcal{B}_k$. We conclude that $\mathcal{A}_k = \bigcup_{i=1,\ldots,5} \mathcal{A}_k^i$ and $\mathcal{B}_k = \mathcal{B}_k^1 \cup \mathcal{B}_k^2$.

From Observation 4.14, Lemmata 4.11 and 4.12, and Relations (4.1)–(4.7) we derive that

$$\begin{aligned}
|\mathcal{A}_k^1| &= \binom{\alpha_{k-1}}{2} \cdot \gamma_{k+1}^2, \\
|\mathcal{A}_k^2| &= \binom{\beta_{k-1}}{2} \cdot \gamma_k^2, \\
|\mathcal{A}_k^3| &= \alpha_{k-1} \cdot \gamma_{k+1} \cdot \beta_{k-1} \cdot \gamma_k, \\
|\mathcal{A}_k^4| &= \alpha_{k-1} \cdot \binom{\gamma_{k+1}}{2} \\
|\mathcal{A}_k^5| &= \beta_{k-1} \cdot \binom{\gamma_k}{2} \\
|\mathcal{B}_k^1| &= \alpha_{k-1} \cdot \gamma_{k+1} \\
|\mathcal{B}_k^2| &= \beta_{k-1} \cdot \gamma_k
\end{aligned}$$

Therefore,

$$\begin{aligned}
\alpha_k &= \binom{\alpha_{k-1}}{2}\gamma_{k+1}^2 + \binom{\beta_{k-1}}{2}\gamma_k^2 + \alpha_{k-1}\binom{\gamma_{k+1}}{2} + \\
&\quad \beta_{k-1}\binom{\gamma_k}{2} + \alpha_{k-1}\beta_{k-1}\gamma_k\gamma_{k+1} \qquad (4.8) \\
\beta_k &= \alpha_{k-1}\gamma_{k+1} + \beta_{k-1}\gamma_k \qquad (4.9)
\end{aligned}$$

By simplifying (4.8),

$$\begin{aligned}
\alpha_k &= \frac{1}{2}\left[\left(\gamma_{k+1}^2\alpha_{k-1}^2 + \gamma_k^2\beta_{k-1}^2 + 2\alpha_{k-1}\beta_{k-1}\gamma_k\gamma_{k+1}\right) - \right. \\
&\quad \left.(\alpha_{k-1}\gamma_{k+1} + \beta_{k-1}\gamma_k)\right] \\
&= \frac{1}{2}\left(\beta_k^2 - \beta_k\right).
\end{aligned}$$

It follows (using Relation (4.9)) that,

$$\tau_k = \frac{1}{2}\left(\beta_k^2 + \beta_k\right) \text{ and } \beta_k = \gamma_k\beta_{k-1}^2$$

Let $\delta_k = 2^{k-1} - k$ and observe that $\beta_k = 2^{\delta_k} = 2^{2^{k-1}-k}$, for every integer $k \geq 2$. Then $\tau_k = 2^{2^k-(2k+1)} + 2^{2^{k-1}-(k+1)}, k \geq 3$, and the theorem follows. $\square$

4.6 Obstructions for $\mathcal{G}_k$, $k \le 3$

It is easy to prove that

- $\mathbf{obs}_{\le_m}(\mathcal{G}_0) = \mathbf{obs}_{\subseteq}(\mathcal{G}_0) = \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_0) = \{K_1\}$,
- $\mathbf{obs}_{\le_m}(\mathcal{G}_1) = \mathbf{obs}_{\subseteq}(\mathcal{G}_1) = \mathbf{obs}_{\sqsubseteq}(\mathcal{G}_1) = \{K_2\}$,
- $\mathbf{obs}_{\le_m}(\mathcal{G}_2) = \mathbf{obs}_{\subseteq}(\mathcal{G}_1) = \{K_3, P_4\}$ and $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_2) = \{K_3, P_4, C_4\}$.

Let $\mathcal{D}$ be the set of the graphs that appear inside the outer polygon in Figure 4.2. In this section we prove that $\mathbf{obs}_{\subseteq}(\mathcal{G}_3) = \mathcal{D}$.

Theorem 4.5. *For any graph G, $\mathbf{td}(G) > 3$ if and only if G contains one of the graphs in $\mathcal{D}$ as a subgraph.*

Proof. Since each of the graphs in $\mathcal{D}$ is connected and has tree-depth four, it suffices to show that any connected graph with tree-depth four contains one of them as a subgraph. Suppose for contradiction that this is not the case, and let G be a connected graph with tree-depth four that contains none of the graphs in $\mathcal{D}$ as a subgraph. We may assume that G is minimal, that is, that $\mathbf{td}(G \backslash e) = 3$ and $\mathbf{td}(G \backslash v) = 3$ for any edge $e \in E(G)$ and any vertex $v \in V(G)$. The graph G cannot contain any cycles of length greater than four, otherwise, it would contain C_5, C_6, C_7, or P_8 as a subgraph.

Let G' be a 2-connected subgraph of G, and suppose that $|V(G')| \geq 5$. Observe that G' contains a 4-cycle $C = v_1v_2v_3v_4$. Consider a vertex $v_5 \in V(G') \setminus V(C)$. Since G' is 2-connected, there exists a path P with distinct endpoints in C such that $v_5 \in V(P)$ and $|V(P) \cap V(C)| = 2$.

Since G does not contain cycles of length at least 5, P has length two and joins two opposite vertices of C, say v_1 and v_3. If the subgraph induced by $V(C) \cup \{v_5\}$ contains any of the edges $\{v_2, v_4\}$, $\{v_2, v_5\}$ or $\{v_4, v_5\}$, then G contains C_5 as a subgraph, hence we may assume that this is not the case. Also, none of v_2, v_4 and v_5 may be incident with any other vertex of G, otherwise G would contain K_4^2. Consider the graph H obtained from G by removing the edge $\{v_1, v_5\}$. By the minimality of G, $\mathbf{td}(H) = 3$. The graph H is connected, hence H contains a vertex v such that $H \setminus v$ is a star forest. If $v = v_1$ or $v = v_3$, then $G \setminus v$ is a star forest, which is contradiction with $\mathbf{td}(G) = 4$. However, $H \setminus v$ for any other vertex v contains P_4 as a subgraph. This is a contradiction, hence we may assume that any 2-connected subgraph of G has at most four vertices.

Let us now consider the case where G contains a 4-cycle $C = v_1v_2v_3v_4$. If both edges $\{v_1, v_3\}$ and $\{v_2, v_4\}$ are in G, then G contains K_4 as a subgraph, thus we may assume this is not the case. Suppose first that $\{v_1, v_3\}$ is an edge (thus $\{v_2, v_4\}$ is not an edge). If v_2 or v_4 is adjacent to a vertex outside of C, then G contains K_4^1 as a subgraph. Otherwise, consider the graph H obtained from G by removing the edge $\{v_1, v_3\}$. By the minimality of G, there exists a vertex v such that $H \setminus v$ is a star forest. The vertex v must belong to C. Since $G \setminus v$ is not a star forest, $v \neq v_1$ and $v \neq v_3$, hence we may assume that $v = v_2$. Since $H \setminus v_2$ is a star forest, v_4 is the only neighbor of v_1 and v_3 in $H \setminus v_2$. But then $H = C$, and tree-depth of G would be only three, which is a contradiction; therefore, any 4-cycle in G is induced.

Let $C = v_1v_2v_3v_4$ be an induced 4-cycle in G. Since G does not contain K_4^2 as a subgraph, the vertices of $V(G) \setminus V(C)$ can only be adjacent to two non-adjacent vertices of C, say v_1 and v_3. Since $\mathbf{td}(G) = 4$, we have $G \neq C$ and we may assume that there exists a vertex $v_5 \in V(G) \setminus V(C)$

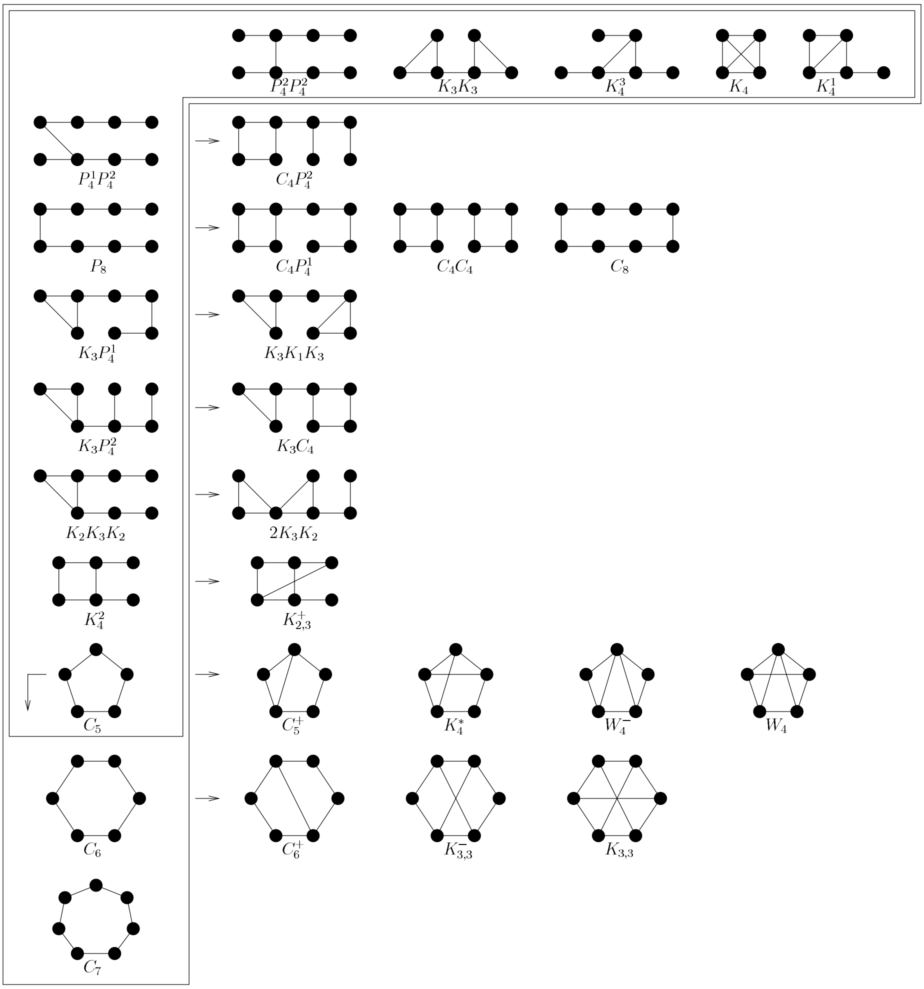


Figure 4.2: The forbidden graphs for $\mathcal{G}_3$.

adjacent to v_1. Let us consider the graph H obtained from G by removing the edge v_1v_4. By the minimality of G, there exists a vertex v such that $H \setminus v$ is a star forest. Since $v_5v_1v_2v_3v_4$ is a path, v must be v_1, v_2, or v_3. If $v = v_1$ or $v = v_3$, then $G \setminus v$ is a star forest, hence $v = v_2$. However, this means that $G \setminus v_1$ is a star forest, which is a contradiction, thus G does not contain any 4-cycle.

Consider now the case where G contains a triangle $C = v_1v_2v_3$. The graph G cannot contain another triangle disjoint from C, since otherwise it would contain $K_3P_4^1$ or K_3K_3 as a subgraph. Together with the fact that each nontrivial 2-connected subgraph of G is a triangle, this implies that all the triangles in G intersect in one vertex. We may assume that there is at least one vertex v_4 not belonging to C adjacent to v_1, and that all triangles in G contain the vertex v_1.

The vertex v_1 is a cut-vertex in G. The graph $G \setminus v_1$ is not a star forest, hence one of its components contains a triangle or P_4. All triangles in G contain the vertex v_1, hence one of the components of $G \setminus v_1$ contains a path P of length three.

If P is disjoint with C, then G contains a subgraph $K_3P_4^1$ or $K_3P_4^2$. It follows that C is the only triangle in G and that the path P intersects $C \setminus v_1$. If the degree of both v_2 and v_3 is greater than two, then G contains the subgraph K_4^3, thus we may assume that degree of v_2 is two and that $P = v_2v_3v_5v_6$ for some vertices v_5 and v_6. Similarly, $G \setminus v_3$ contains P_4 as a subgraph, hence we may assume that there is a vertex v_7 adjacent to v_4. However, the graph G then would contain $K_2K_3K_2$ as a subgraph. Therefore, G does not contain a triangle, and it must be a tree.

It is however easy to verify using Theorem 4.3 that the only tree-depth critical trees with tree-depth four are P_8, $P_4^1P_4^2$ and $P_4^2P_4^2$. It follows that any graph with $\mathbf{td}(G) > 3$ contains one of the graphs in $\mathcal{D}$ as a subgraph. $\square$

Corollary 4.3. *The set* $\mathbf{obs}_{\leq_m}(\mathcal{G}_3)$ *contains exactly all the graphs depicted in the inner polygon in Figure 4.2.*

Proof. Follows directly from Observation 4.2 and the fact that $C_5 \leq_m C_6$ and $C_5 \leq_m C_7$. □

Corollary 4.4. *The set* $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_3)$ *contains exactly all the graphs in Figure 4.2.*

Proof. Follows by inspection, using Observation 4.3. □

Notice that the obstructions for $\mathcal{G}_k$ have at most 2^k vertices for $k \in [3]$. Hence Theorem 4.2 is not sharp even in this case (it only claims that the obstructions have at most 16 vertices). We conclude this section with the following conjecture.

Conjecture 2. *For every* $k \geq 1$*, the order of the graphs in* $\mathbf{obs}_{\sqsubseteq}(\mathcal{G}_k)$ *is bounded by* 2^k*.*

4.7 A Reduction for Tree-depth

In this section, towards the effort of trying to identify minimal graphs that do not belong to the graph class $\mathcal{G}_k$, we prove a reduction which facilitates the identification of these graphs with respect to relations that allow vertex removals.

In particular, we suggest a procedure which permits us to remove a vertex set from a graph G while preserving its tree-depth.

Given a graph G we say that a set $S \subseteq V(G)$ is *a set of siblings* if for every $x, y \in S$, $N_G(x) = N_G(y)$. Consider the following.

Observation 4.16. *Let* G *be a graph and* ρ *be a* k*-vertex ranking of* G*. Let also* $v_1, v_2 \in V(G)$ *such that* $\{v_1, v_2\} \in E(G)$ *and* $\rho(v_1) < \rho(v_2)$*. Then* $\rho(v_2) \notin \rho(N_{G \setminus \{v_2\}}(v_1))$*.*

We now prove the following general reduction-lemma.

Lemma 4.16. *Let k be a positive integer, G be a graph and $S \subseteq V(G)$ be a set of siblings of G each of degree k. Let also $G' = G \setminus S'$ where S' is any subset of S such that $|S'| \leq |S| - k$. Then $\mathbf{td}(G) = \mathbf{td}(G')$.*

Proof. We examine the non-trivial case where $|S| \geq k+1$. We denote $S'' = S \setminus S' = \{u_i \mid i \in |S''|\}$. As G' is a subgraph of G, it is enough to prove that $\mathbf{td}(G) \leq \mathbf{td}(G')$. Let $\rho' : V(G') \to \{1, \ldots, t\}$ be a vertex ranking of G'. Let $N = \{v_i \mid i \in [k]\}$ be the common neighbourhood of the vertices in S'' and without loss of generality assume that $\rho'(v_i) \leq \rho'(v_{i+1}), i \in [k-1]$. Notice that $|S''| \geq k$ and without loss of generality assume that $\rho'(u_i) \leq \rho'(u_{i+1}), i \in [|S''| - 1]$. We need the following claim.

Claim 4. *Let P be a (z', z)-path in G where $z \in S''$, $z' \in (G \setminus S'') \setminus N$, and $\rho'(z) = \rho'(z')$. Let P' be the portion of P between z' and the first vertex, say x, in N (recall that N is a separator of G). Then there exists a vertex $y \in V(P') \setminus \{z'\}$ such that $\rho'(y) > \rho'(z')$.*

Proof of Claim 4. It is enough to observe that the path $P'' = (V(P') \cup \{z\}, E(P') \cup \{\{x, z\}\})$ should contain an internal vertex y where $\rho'(y) > \rho'(z')$. □

In what follows we construct a vertex ranking $\rho : V(G) \to \{1, \ldots, t\}$. Let

$$m = \begin{cases} \max\{i \mid \rho'(u_1) > \rho'(v_i)\} + 1 & \text{if } A = \{i \mid \rho'(u_1) > \rho'(v_i)\} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

and observe that $m \leq k+1$. We claim that

$$\rho = \{(x, \rho'(x)) \mid x \in V(G') \setminus (S'' \cup \bigcup_{i \in [m-1]} \{v_i\})\} \cup \sigma$$

where

$$\sigma = \begin{cases} \{(v_i, \rho'(u_{i+1})) \mid i \in [m-1]\} \cup \{(x, \rho'(u_1)) \mid x \in S\} & m \neq k+1 \\ \{(v_i, \rho'(u_i)) \mid i \in [m-1]\} \cup \{(x, \rho'(v_1)) \mid x \in S\} & m = k+1 \end{cases}$$

is a t-vertex ranking of G.

First we examine the case where $m = 1$. Then observe that

$$\rho'' = \{(x, \rho'(x)) \mid x \in V(G') \setminus S''\} \cup \{(x, \rho'(u_1)) \mid x \in S''\}$$

is a t-vertex ranking of G'. It is easy to observe that $\rho'' \cup \{(x, \rho'(u_1)) \mid x \in S'\}$ is a t-vertex ranking of G that is equal to ρ.

We examine now the case where $1 < m \leq k+1$. As $A \neq \emptyset$, Observation 4.16 implies that

$$\rho'(u_i) < \rho'(u_{i+1}), \qquad i \in [|S''| - 1] \tag{4.10}$$

$$\rho'(v_i) < \rho'(v_{i+1}), \qquad m \leq i \leq k-1 \tag{4.11}$$

$$\rho'(N_{G' \setminus S''}(\bigcup_{i \in [m-1]} \{v_i\})) \cap \rho'(S'') = \emptyset \tag{4.12}$$

thus, from (4.10), $|\rho'(S'')| = |S''| \geq k$. We distinguish the following cases:

Case 1. $1 < m < k+1$. We claim that

$$\begin{aligned} \rho'' \quad = \quad & \{(x, \rho'(x)) \mid x \in V(G') \setminus (S'' \cup \bigcup_{i \in [m-1]} \{v_i\})\} \cup \\ & \{(v_i, \rho'(u_{i+1})) \mid i \in [m-1]\} \cup \{(x, \rho'(u_1)) \mid x \in S''\} \end{aligned}$$

is a t-vertex ranking of G'. Indeed, ρ'' is a proper coloring of G' because of (4.10), (4.11), and (4.12). To prove that ρ'' is a t-vertex ranking, we consider a (z', z)-path P between two vertices $z, z' \in V(G')$ where $\rho''(z) = \rho''(z')$. We observe the following.

Claim 5. $|\rho''(N)| = k$.

Proof of Claim 5. It follows directly from (4.10) and (4.11). □

We distinguish the following subcases.

Subcase 1.1. If one, say z, of the endpoints of P belongs to S'', then P contains at least one vertex $v_i, i \in N$. If $i \in A$ then $\rho''(v_i) \geq \rho'(u_2) > \rho'(u_1) = \rho''(z)$. If $i \in [k] \setminus A$, then $\rho''(v_i) = \rho'(v_i) > \rho'(u_1) = \rho''(z)$.

Subcase 1.2. If one, say z, of the endpoints of P belongs to $N' = \{v_i \mid i \in A\}$, then we assume that $z = v_i$ and, from Claim 5, $z' \in (V(G') \setminus S'') \setminus N$. Let P' be the portion of P between z' and the first vertex x in N. Then from Claim 4, there exists a vertex $y \in V(P') \setminus \{z'\}$ where $\rho'(y) > \rho'(z')$. Observe that $\rho'(z') = \rho''(z')$ and $\rho''(y) \geq \rho'(y)$. Therefore, $\rho''(y) > \rho''(z')$ and we are done as $y \in V(P') \subseteq V(P)$.

Subcase 1.3. If one, say z, of the endpoints of P belongs to $N \setminus N'$, then again from Claim 5, $z' \in (V(G') \setminus S'') \setminus N$. Let P' be the portion of P between z' and the first vertex x in N. If $w = z$ then $P' = P$ and we are done. If $x \neq z$, we define $P'' = (V(P') \cup \{u_1, z\}, E(P) \cup \{\{x, u_1\}, \{u_1, z\}\})$ and observe that $\rho'(z) = \rho''(z')$ and $\rho'(z') = \rho''(z')$. Therefore, P'' contains some internal vertex y where $\rho'(y) > \rho'(z) = \rho''(z)$. Notice also that $\rho'(u_1) < \rho'(z)$, thus $y \in V(P')$. It also holds that $\rho''(y) \geq \rho'(y)$, therefore $\rho''(y) > \rho''(z)$ and we are done as $y \in V(P') \subseteq V(P)$.

Subcase 1.4. If both z, z' belong in $(V(G') \setminus S'') \setminus N$, then we examine the non-trivial case where $V(P) \cap S'' \neq \emptyset$ (recall that the new coloring, only increases the colors not in S''). Let P' (respectively P'') be the portion of P between z (respectively z') and the first vertex x (respectively x') in

N. We define the path $P''' = P' \cup P'' \cup (\{u_1, x, x'\}, \{\{x, u_1\}, \{x', u_1\}\})$. Again $\rho'(z) = \rho''(z')$ and $\rho'(z') = \rho''(z')$ and let y be a vertex in P''' where $\rho'(y) > \rho'(z) = \rho''(z)$. If $y \in V(P') \cup V(P'')$ then we are done as $\rho''(y) \geq \rho'(y)$ and $V(P') \cup V(P'') \subseteq V(P)$. If $y = u_1$, then we are also done as $S'' \cap V(P) \neq \emptyset$ and the color assigned by ρ'' to every vertex in $S'' \cap V(P) \neq \emptyset$ is equal to $\rho'(u_1)$.

We just proved that ρ'' is a t-vertex ranking of G'. It remains now to observe that $\rho'' \cup \{(x, \rho'(u_1)) \mid x \in S'\}$ is a t-vertex ranking of G that is equal to ρ.

Case 2. $m = k + 1$. We claim that

$$\begin{aligned}\rho'' &= \{(x, \rho'(x)) \mid x \in V(G') \setminus (\bigcup_{i \in [m-1]} \{v_i\} \cup S'')\} \cup \\ &\quad \{(v_i, \rho'(u_i)) \mid i \in [m-1]\} \cup \{(x, \rho'(v_1)) \mid x \in S''\}\end{aligned}$$

is a t-vertex ranking of G'.

Observe first that Claim 5 is again true from (4.10).

We distinguish the following subcases.

Subcase 2.1. If one, say z, of the endpoints of P belongs to S'', then P contains at least one vertex $v_i, i \in N$. Then $\rho''(v_i) \geq \rho'(u_1) > \rho'(v_1) = \rho''(z)$.

Subcase 2.2. If one, say z, of the endpoints of P belongs to N, then we assume that $z = v_i$ and, from Claim 5, $z' \in (V(G') \setminus S'') \setminus N$. Let P' be the portion of P between z' and the first vertex x in N. Then from the Claim 4, there exists a vertex $y \in V(P') \setminus \{z'\}$ where $\rho'(y) > \rho'(z')$. Observe that $\rho'(z') = \rho''(z')$ and $\rho''(y) \geq \rho'(y)$. Therefore, $\rho''(y) > \rho''(z')$

and we are done as $y \in V(P') \subseteq V(P)$.

Subcase 2.3. If both z, z' belong to $(V(G') \setminus S'') \setminus N$, then we examine the non-trivial case where $V(P) \cap S'' \neq \emptyset$ (recall that the new coloring, only increases the colors not in S''). Let P' (respectively P'') be the portion of P between z (respectively z') and the first vertex x (respectively x') in N. We define the path $P''' = P' \cup P'' \cup (\{u_1, x, x'\}, \{\{x, u_1\}, \{x', u_1\}\})$. Again $\rho'(z) = \rho''(z')$ and $\rho'(z') = \rho''(z')$ and let y be a vertex in P''' where $\rho'(y) > \rho'(z) = \rho''(z)$. If $y \in V(P') \cup V(P'')$ then we are done as $\rho''(y) \geq \rho'(y)$ and $V(P') \cup V(P'') \subseteq V(P)$. If $y = u_1$, then we are done as $\rho''(x) \geq \rho'(u_1)$ and $x \in V(P') \cup V(P'') \subseteq V(P)$.

We just proved that ρ'' is a t-vertex ranking of G'. It remains to observe that $\rho'' \cup \{(x, \rho'(v_1)) \mid x \in S'\}$ is a t-vertex ranking of G that is equal to ρ. □

CHAPTER 5

Computing Immersion Obstructions

The development of the Graph Minor Theory constitutes a vital part of modern Combinatorics. A lot of theorems that were proved and techniques that were introduced in its context appear to be of crucial importance in Algorithmics and Parameterized Complexity Theory as well as in Structural Graph Theory. Such examples are the Excluded Grid Theorem [197], the Structural Theorems in [193, 207] and the Irrelevant Vertex Technique in [185]. (For examples of algorithmic applications, see [50, 109] and Chapter 8).

However, while the minor ordering has been extensively studied throughout the last decades [4, 29, 185, 193, 197, 207, 210, 211], the immersion ordering has only recently gained more attention [58, 109, 216]. Recall that one of the fundamental results that appeared in the last paper of the Graph Minors series was the proof of Nash-Williams' Conjecture (see

Section 3.1), that is, the class of all graphs is well-quasi-ordered by the immersion ordering [211].

As we have already seen, a direct corollary of these results is that a graph class $\mathcal{C}$, which is closed under taking immersions, can be characterized by a finite family $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ of minimal, according to the immersion ordering, graphs that are not contained in $\mathcal{C}$. Furthermore, in [109], it was proven that there is an $O(|V(G)|^3)$ algorithm that decides whether a graph H is an immersion of a graph G (where the hidden constants depend only on H). Thus, an immediate algorithmic implication of the finiteness of $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ and the algorithm in [109], is that it can be decided in cubic time whether a graph belongs to $\mathcal{C}$ or not (by testing if the graph G contains any of the graphs in $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ as an immersion). In other words, these two results imply that membership in an immersion-closed graph class can be decided in cubic time. (See also Section 3.3)

Recall here that the same meta-algorithmic conclusion holds for the minor ordering from the proofs in [193] and [194]. Evenmore, this result, that is, the existence of a cubic time algorithm deciding the membership of a graph in a graph class that is closed under minors, broadened the perspectives towards the understanding of the NP-hard problems. It was actually at that point, according to M. Langston, that it became clear what seemed to be as "*different levels of hardness*" between these problems [22]. Notice for example, for the well-known k-Vertex Cover problem, that the class of graphs admitting a vertex cover of size at most k is closed under taking of minors. Therefore, for every fixed k there is a cubic time algorithm deciding whether a graph has a vertex cover of size k. However, no similar result can be expected for the k-Coloring problem, as it is known to be NP-hard for every fixed $k \geq 3$. The observation of this gap in the time complexity of the NP-hard problems encouraged the development of the Parameterized Complexity Theory [66, 82, 172] by M. Fellows and R. Downey (see also Chapter 8). This theory has proven to be very powerful and has majorly advanced during the past decades (for example, see [10, 23, 24, 41, 50]).

Nevertheless, recall that, the aforementioned meta-algorithmic result for an immersion-closed graph class $\mathcal{C}$ assumes that the family $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ is known. Evenmore, as the proofs in [194] and [211] are non-constructive (see [95]), no generic algorithm is provided that allows us to identify these obstruction sets for every immersion-closed graph class. Moreover, even for fixed graph classes, this task can be extremely challenging as such a set could contain many graphs and no general upper bound on its cardinality is known other than its finiteness [69] (see, also, the previous chapter).

The issue of the computability of obstruction sets for minors and immersions was raised by M. Fellows and M. Langston [78, 79] and the challenges towards computing obstruction sets soon became clear. In particular, in [79], M. Fellows and M. Langston showed that the problem of determining obstruction sets from machine descriptions of minor-closed graph classes is recursively unsolvable (which directly holds for the immersion ordering as well). Evenmore, in [37], B. Courcelle, R. Downey and M. Fellows proved that the obstruction set of a minor-closed graph class cannot be computed from a description of the minor-closed graph class in Monadic Second Order Logic (MSO). Thus, a consequent open problem is to identify the information that is needed for an immersion-closed graph class $\mathcal{C}$ in order to make it possible to effectively compute the obstruction set $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$.

Several methods have been proposed towards tackling the non constructiveness of these sets (see, for example, [29, 78]) and the problem of algorithmically identifying minor obstruction sets has been extensively studied [4, 29, 37, 78, 79, 144]. In [29], it was proven that the obstruction set of a minor-closed graph class $\mathcal{F}$ which is the union of two minor-closed graph classes $\mathcal{F}_1$ and $\mathcal{F}_2$ whose obstruction sets are given can be computed under the assumption that there is at least one tree that does not belong to $\mathcal{F}_1 \cap \mathcal{F}_2$. Evenmore, in [4], it was shown that the aforementioned assumption is not necessary.

In this section, we initiate the study for computing immersion obstruction sets. In particular, we deal with the problem of computing the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ for families of graph classes $\mathcal{C}$ that are constructed by finite unions of immersion-closed graph classes. Notice that the union and the intersection of two immersion-closed graph classes are also immersion-closed, hence their obstruction sets are of finite size. It is also easy to see that, given the obstruction sets of two immersion-closed graph classes, the obstruction set of their intersection can be computed in a trivial way. We prove that there is an algorithm that, given the obstruction sets of two immersion-closed graph classes, outputs the obstruction set of their union.

Our approach is based on the derivation of an upper bound on the treewidth of the obstructions of an immersion-closed graph class. Notice that the combination of a machine description of an immersion-closed graph class $\mathcal{F}$ with an upper bound on the size of the forbidden graphs makes this computation possible, but neither the machine description of the graph class nor the upper bound alone are sufficient information. Moreover, as mentioned before, no generic procedure is known for computing such an upper bound. We build on the machinery introduced by I. Adler, M. Grohe and S. Kreutzer in [4] for computing minor obstruction sets. In particular, we will ask for an MSO-description of an immersion-closed graph class instead of a machine description, and a bound on the treewidth instead of an upper bound on the size of the obstructions of the immersion-closed graph class.

For this, we adapt the results on [4] so to permit the computation of the obstruction set of any immersion-closed graph class, under the conditions that an explicit upper bound on the tree-width of its obstructions can also be computed and the class can be defined in MSO. We present this algorithm at Lemma 5.4, and with that we conclude the computability part of the chapter. Our next step is a combinatorial result proving an upper bound on the tree-width of the obstructions of the union of two immersion-closed graph classes, whose obstruction sets are known. We then show that the obstruction set of their union can be effectively computed. Our combinatorial proofs significantly differ from the ones in [4] and make use of a suitable extension of the Unique Linkage Theorem of K. Kawarabayashi and P. Wollan [124].

The remainder of this chapter is structured as follows. In Section 5.1 we state the basic notions that we use throughout the section as well as few well-known results. In Section 5.2 we present our computability result, that is, we prove that the obstruction set of an immersion-closed graph class can be computed when an upper bound on the tree-width of its obstructions and an MSO-description of the graph class are known. We do so by proving a version of Lemma 3.1 of [4], adapted to the immersion ordering. In Section 5.3 we provide the bounds on the tree-width of the graphs that belong to the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1 \cup \mathcal{C}_2)$ by assuming that the sets $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1)$ and $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_2)$ are known, where $\mathcal{C}_1$ and $\mathcal{C}_2$ are immersion-closed graph classes. By doing this we propagate the computability of immersion obstruction sets to finite unions of immersion-closed graph classes.

5.1 Preliminaries

We define an ordering $\leq$ between finite sets of graphs as follows: $\mathcal{F}_1 \leq \mathcal{F}_2$ if and only if

1. $\sum_{G\in\mathcal{F}_1} |V(G)| < \sum_{H\in\mathcal{F}_2} |V(H)|$ or
2. $\sum_{G\in\mathcal{F}_1} |V(G)| = \sum_{H\in\mathcal{F}_2} |V(H)|$ and $\sum_{G\in\mathcal{F}_1} |E(G)| < \sum_{H\in\mathcal{F}_2} |E(H)|$.

Definition 5.1 (An equivalent definition of obstruction sets.)**.** Let $\mathcal{C}$ be an immersion-closed graph class. A set of graphs $F = \{H_1, \ldots, H_n\}$ is called (immersion) obstruction set of $\mathcal{C}$, and is denoted by $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$, if and only if F is a $\leq$-minimal set of graphs for which the following holds: For every graph G, G does not belong to $\mathcal{C}$ if and only if there exists a graph $H \in F$ such that $H \leq_{im} G$.

Remark 2. While we have already defined obstructions sets we also wish to include Definition 5.1 as it may facilitate the understanding of the proof of Lemma 5.4.

Recall that, because of the seminal result of N. Robertson and P. Seymour [211], for every immersion-closed graph class $\mathcal{C}$, the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ is finite.

In [210], N. Robertson and P. Seymour proved a theorem which is known as The Vital Linkage Theorem. This theorem provides an upper bound for the tree-width of a graph G that contains a vital k-linkage L such that $V(L) = V(G)$, where the bound depends only on k. A stronger statement of the Vital Linkage Theorem was recently proved by K. Kawarabayashi and P. Wollan [124], where instead of asking for the linkage to be vital, it asks for it to be unique. Notice here that a vital linkage is also unique. As in some of our proofs (for example, the proof of Lemma 5.5) we deal with unique but not necessarily vital linkages we make use of the Vital Linkage Theorem in its latter form which is stated below.

Theorem 5.1 (The Unique Linkage Theorem [124, 210])**.** *There exists a computable function $w : \mathbb{N} \to \mathbb{N}$ such that the following holds. Let L be a (1-approximate) k-linkage in G with $V(L) = V(G)$. If L is unique then $\mathbf{tw}(G) \leq w(k)$.*

Monadic Second Order Logic

Lemma 5.1. *The class of graphs that contain a fixed graph H as an immersion is MSO-definable by an MSO-formula ϕ_H.*

Proof. Let $V(H) = \{v_1, v_2, \ldots, v_n\}$ and $E(H) = \{e_1, e_2, \ldots, e_m\}$. Let also ϕ_H be the following formula.

$$\begin{aligned}\phi_H :=& \exists E_1, E_2, \ldots, E_m \exists x_1, x_2, \ldots, x_n \Big[(\bigwedge_{i \in [n]} V(x_i)) \wedge (\bigwedge_{j \in [m]} E_i \subseteq E) \wedge \\ & (\bigwedge_{i \neq j} x_i \neq x_j) \wedge (\bigwedge_{p \neq q} E_p \cap E_q = \emptyset) \wedge (\bigwedge_{e_r = \{v_k, v_l\} \in E(H)} \mathrm{path}(x_k, x_l, E_r)) \Big],\end{aligned}$$

where $\mathrm{path}(x, y, Z)$ is the MSO formula stating that the edges in Z form a path from x to y. This can be done by saying that the set Z of edges is connected and every vertex v incident to an edge in Z is either incident to exactly two edges of Z or to exactly one edge with further condition that $v = x$ or $v = z$. Thus, $\mathrm{path}(x, y, Z)$ can be expressed in MSO by the following formula.

$$
\begin{aligned}
&[(x \neq y) \wedge \exists p, q(Z(p) \wedge Z(q) \wedge I(x,p) \wedge I(y,q) \wedge \\
&\forall p' \in Z(I(x,p') \rightarrow p = p') \wedge \forall q' \in Z(I(y,q') \rightarrow q = q')) \wedge \\
&\forall w(V(w) \wedge w \neq x \wedge w \neq y \wedge \exists q_1(Z(q_1) \wedge I(w,q_1)) \rightarrow \\
&\exists q_2, q_3(Z(q_2) \wedge Z(q_3) \wedge q_2 \neq q_3 \wedge I(w,q_2) \wedge I(w,q_3))) \wedge \\
&\forall p_1, p_2, p_3(Z(p_1) \wedge Z(p_2) \wedge Z(p_3) \wedge \\
&\exists m(V(m) \wedge I(u,p_1) \wedge I(u,p_2) \wedge I(u,p_3)) \rightarrow \bigvee_{i \neq j} (p_i = p_j))]
\end{aligned}
$$

It is easy to verify that ϕ_H is the desired formula. □

We now state a theorem which plays a crucial role in the proof of our algorithm for the computation of immersion obstructions for general immersion-closed graph classes.

Theorem 5.2 ([13, 36])**.** *Let ϕ be a fixed MSO-formula. There is an algorithm such that, for every positive integer k and every graph G, whose tree-width is upper bounded by k and is given together with a tree-decomposition, decides where ϕ is satisfied by the graph G .*

In [4], I. Adler, M. Grohe and S. Kreutzer provide tools that allow us to use Theorem 5.2, when an upper bound on the tree-width of the obstructions is known and an MSO-description of the graph class can be computed, in order to compute the obstruction sets of minor-closed graph classes. We adapt their machinery to the immersion ordering and prove that the tree-width of the obstructions of immersion-closed graph classes is upper bounded by some function that only depends on the graph class. This provides a generic technique to construct immersion obstruction sets when the explicit value of the function is known. Then, by obtaining such a computable upper bound on the tree-width of the graphs in $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$, where $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}_1, \mathcal{C}_2$ are immersion-closed graph classes whose obstruction sets are given, we show that the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ can be effectively computed.

5.2 Computing Immersion Obstruction Sets

In this section we prove the analogue of Lemma 2.2 in [4] (Lemma 5.2) and the analogue of Lemma 3.1 in [4] (Lemma 5.4) for the immersion ordering.

We first state the combinatorial lemma of this section.

Lemma 5.2. *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that the following holds. Let H and G be graphs such that $H \leq_{im} G$. If G' is a minimal subgraph of G with $H \leq_{im} G'$ then $\mathbf{tw}(G') \leq f(|E(H)|)$.*

The proof of Lemma 5.2 is omitted as a stronger statement will be proved later on (Lemma 5.7). We continue by giving the necessary definitions in order to prove the analogue of Lemma 3.1 in [4] for the immersion ordering.

Extension of MSO For convenience, we consider the extension of the signature $\tau_{\mathcal{G}}$ to a signature τ_{ex} that pairs the representation of a graph G with the representation of one of its tree-decompositions.

Definition 5.2. If G is a graph and $\mathcal{T} = (\mathcal{X}, T)$ is a tree-decomposition of G, τ_{ex} is the signature that consists of the relation symbols V, E, I of $\tau_{\mathcal{G}}$, and four more relation symbols V_T, E_T, I_T and X.
A *tree-dec expansion* of G and $\mathcal{T}$, is a τ_{ex}-structure

$$\begin{aligned}\mathfrak{G}_{ex} = (&V(G) \cup E(G) \cup V(T) \cup E(T),\\ &V^{\mathfrak{G}_{ex}}, E^{\mathfrak{G}_{ex}}, I^{\mathfrak{G}_{ex}}, V_T^{\mathfrak{G}_{ex}}, E_T^{\mathfrak{G}_{ex}}, I_T^{\mathfrak{G}_{ex}}, X^{\mathfrak{G}_{ex}})\end{aligned}$$

where $V_T^{\mathfrak{G}_{ex}} = V(T)$ represents the node set of T, $E_T^{\mathfrak{G}_{ex}} = E(T)$ the edge set of T, $I_T^{\mathfrak{G}_{ex}} = \{(v,e) \mid v \in e \cap V(T) \wedge e \in E(T)\}$ the incidence relation in T and $X^{\mathfrak{G}_{ex}} = \{(t,v) \mid t \in V(T) \wedge v \in X_t \cap V(G)\}$.

We denote by $\mathcal{C}_{\mathcal{T}_k}$ the class of tree-dec expansions consisting of a graph G with $\mathbf{tw}(G) \leq k$, and a tree decomposition $(\mathcal{X}, T)$ of G of width$(\mathcal{X}, T) \leq k$.

Lemma 5.3 ([4])**.** *It holds that:*

1. *If G is a graph and $(\mathcal{X}, T)$ is a tree decomposition of it whose width is at most k then the tree-width of the tree-dec expansion of G is at most $k+2$.*
2. *There is an MSO-sentence $\phi_{\mathcal{C}_{\mathcal{T}_k}}$ such that for every τ_{ex}-structure $\mathfrak{G}$, $\mathfrak{G} \models \phi_{\mathcal{C}_{\mathcal{T}_k}}$ if and only if $\mathfrak{G} \in \mathcal{C}_{\mathcal{T}_k}$.*

A classic result [13] (see Theorem 5.2) states that we can decide, for every $k \geq 0$, if an MSO-formula is satisfied in a graph G of $\mathbf{tw}(G) \leq k$. An immediate corollary of this result and Lemma 5.3 is the following.

Corollary 5.1. *We can decide, for every k, if an MSO-formula ϕ is satisfied in some $\mathfrak{G} \in \mathcal{C}_{\mathcal{T}_k}$.*

Theorem 5.3 ([4])**.** *For every $k \geq 0$, there is an MSO-sentence $\phi_{\mathcal{T}_k}$ such that for every tree-dec expansion $\mathfrak{G} \in \mathcal{C}_{\mathcal{T}_l}$ of G, for some $l \geq k$, it holds that $\mathfrak{G} \models \phi_{\mathcal{T}_k}$ if and only if $\mathbf{tw}(G) = k$.*

Definition 5.3. A graph class $\mathcal{C}$ is *layer-wise MSO-definable*, if for every $k \in \mathbb{N}$ we can compute an MSO-formula ϕ_k such that $G \in \mathcal{C} \wedge \mathbf{tw}(G) \leq k$ if and only if $\mathfrak{G} \models \phi_k$, where $\mathfrak{G} \in \mathcal{C}_{\mathcal{T}_k}$ is the tree-dec expansion of G.

Definition 5.4. Let $\mathcal{C}$ be an immersion-closed graph class. The *width* of $\mathcal{C}$, $\mathbf{width}(\mathcal{C})$ is the minimum positive integer k such that for every graph $G \notin \mathcal{C}$ there is a graph $G' \subseteq G$ with $G' \notin \mathcal{C}$ and $\mathbf{tw}(G') \leq k$.

Note that Lemma 5.2 ensures that the width of an immersion-closed graph class is well-defined.

Observation 5.1. *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are immersion-closed graph classes then the following hold.*

1. *For every graph $G \notin \mathcal{C}_1 \cup \mathcal{C}_2$, there exists a graph $G' \subseteq G$ such that $G' \notin \mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathbf{tw}(G') \leq \max\{r(|E(H)|, |E(J)|) \mid H \in \mathbf{obs}_{\leq_{im}}(\mathcal{C}_1), J \in \mathbf{obs}_{\leq_{im}}(\mathcal{C}_2)\}$, where r is the function of Lemma 5.7 and thus,*

2. *For every graph $G \notin \mathcal{C}_1$, there exists a graph $G' \subseteq G$ such that $G' \notin \mathcal{C}_1$ and $\mathbf{tw}(G') \leq \max\{f(|E(H)|) \mid H \in \mathbf{obs}_{\leq_{im}}(\mathcal{C}_1)\}$, where f is the function of Lemma 5.2.*

Finally, we state the analogue of Lemma 3.1 in [4] for the immersion ordering.

Lemma 5.4. *There exists an algorithm that, given an upper bound $l \geq 0$ on the width of a layer-wise MSO-definable class $\mathcal{C}$, and a computable function $f : \mathbb{N} \to MSO$ such that for every positive integer k, $f(k) = \phi_k$, where ϕ_k is the MSO-formula defining $\mathcal{C} \cap \mathcal{T}_k$, it computes $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$.*

Proof. In order to prove the Lemma it is enough to prove the following.

Claim 6. *For any finite family of graphs $\mathcal{F} = \{F_1, \ldots, F_n\}$, it is decidable whether the following two following conditions are unsatisfiable for a given graph G.*

1. *$G \in C$ and there exists an $F \in \mathcal{F}$ such that $F \leq_{im} G$.*

2. *$G \notin \mathcal{C}$ and for every $F \in \mathcal{F}$, $F \not\leq_{im} G$.*

To see that the above Claim is enough, first notice that if $\mathcal{F}$ is a finite family of graphs for which the formulas χ and ψ are unsatisfiable then $\mathcal{F}$ is a forbidden immersion characterization of $\mathcal{C}$, that is, a graph G belongs to $\mathcal{C}$ if and only if it does not contain any of the graphs in $\mathcal{F}$ as an immersion. By definition, $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ is the minimum such family according to the relation $\leq$ defined in Section 5.1. Thus, if Claim 6 holds, we can find the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ by enumerating, according to $\leq$, all the finite families of graphs $\mathcal{F}$ and deciding, for each one of them, if the formulas χ and ψ are unsatisfiable.

Proof of Claim 6. Let G be a graph in $\mathcal{C}$ such that $F \leq_{im} G$, for some $F \in \mathcal{F}$. Lemma 5.2 implies that there exists a graph $G' \subseteq G$ such that $\mathbf{tw}(G') \leq f(|E(F)|)$ and $F \leq_{im} G'$, where f is the function of Lemma 5.2. Observe that $G' \in \mathcal{C}$. Thus, χ is satisfiable if and only if there exists a graph in $\mathcal{C}$, whose tree-width is bounded from $\max\{f(|E(F)|) : F \in \mathcal{F}\}$, that satisfies it, where f is the computable function of Lemma 5.2. Let $\phi_{\mathcal{C}}$ be the formula defining $\mathcal{C} \cap \mathcal{T}_k$ in $\mathcal{C}_{\mathcal{T}_k}$, and $\phi_{\mathcal{F}} \equiv \bigvee_{F \in \mathcal{F}} \phi_F$, where ϕ_F is the formula from Lemma 5.1 and $k = \max\{f(|E(F)|) : F \in \mathcal{F}\}$. Notice that there exists some graph $G \in \mathcal{C}$ that models $\phi_{\mathcal{F}}$ if and only if $\phi_{\mathcal{C}} \wedge \phi_{\mathcal{F}}$ is satisfiable for some $G' \in \mathcal{C}_{\mathcal{T}_k}$. From Corollary 5.1, this is decidable.

Let $G \notin \mathcal{C}$ be a graph such that $F \nleq_{im} G$, for every $F \in \mathcal{F}$. Recall that the width of a graph class $\mathcal{C}$ is the minimum positive integer k such that for every graph $G \notin \mathcal{C}$ there is a $G' \subseteq G$ with $G' \notin \mathcal{C}$ and $\mathbf{tw}(G') \leq k$. Thus, G contains a subgraph G' with tree-width at most w such that $G' \notin \mathcal{C}$, where w is computable by Lemma 5.2. Observe that $F \nleq_{im} G'$, for every $F \in \mathcal{F}$. If $\phi'_{\mathcal{C}}$ is the MSO-sentence defining $\mathcal{C} \cap \mathcal{T}_w$ (given by the hypothesis), then there exists a graph $G \notin \mathcal{C}$ such that $F \nleq_{im} G$, for every $F \in \mathcal{F}$ if and only if $\neg\phi'_{\mathcal{C}} \wedge \neg\phi_{\mathcal{F}}$ is satisfiable in $\mathcal{C}_{\mathcal{T}_w}$. The decidability of whether $\neg\phi'_{\mathcal{C}} \wedge \neg\phi_{\mathcal{F}}$ is satisfiable in $\mathcal{C}_{\mathcal{T}_w}$ follows, again, from Corollary 5.1. $\square$

As Claim 6 holds, the lemma follows. □

Corollary 5.2. *There is an algorithm that given an MSO formula ϕ and $k \in \mathbb{N}$, so that ϕ defines an immersion closed-graph class $\mathcal{C}$ of width at most k, computes the obstruction set of $\mathcal{C}$.*

We would like to remark here that while Lemma 5.4 states the necessary conditions for the computation of the immersion instruction set for any immersion-closed graph class, this result is generic and there is no uniform way for computing either an upper bound on the width of $\mathcal{C}$ or an MSO-description of $\mathcal{C}$.

In what follows, by proving some combinatorial lemmata, we are able to conclude that if $\mathcal{C}_1$ and $\mathcal{C}_2$ are two immersion-closed graph classes, whose obstruction sets are known, then the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1 \cup \mathcal{C}_2)$ is computable.

5.3 Tree-width Bounds for the Obstructions

In this section, we give an upper bound on the tree-width of the immersion obstructions of the graph class $\mathcal{C}_1 \cup \mathcal{C}_2$ where $\mathcal{C}_1$ and $\mathcal{C}_2$ are immersion-closed graph classes, given that their obstruction sets are known. In order to do this, we first prove a generalization of the Unique Linkage Theorem. Then we introduce the notion of an r-approximate edge-linkage and work on the minimal graphs not belonging to $\mathcal{C}_1 \cup \mathcal{C}_2$.

Finally, as it is trivial to compute an MSO-description of $\mathcal{C}_1 \cup \mathcal{C}_2$ when we are given the sets $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1)$ and $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_2)$, we show that the obstruction set of $\mathcal{C}_1 \cup \mathcal{C}_2$ is computable.

Lemma 5.5. *There exists a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that the following holds. Let G be a graph that contains a 2-approximate k-linkage $\tilde{L}$ such that $V(\tilde{L}) = V(G)$. If $\tilde{L}$ is unique, then $\mathbf{tw}(G) \leq f(k)$.*

Proof. Let G be a graph that contains a unique 2-approximate k-linkage $\tilde{L}$ with $V(\tilde{L}) = V(G)$ that links the sets $A = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $B = (\beta_1, \beta_2, \ldots, \beta_k)$ in G. Denote by T the set $A \cup B$ and consider the graph G^b with

$$\begin{aligned} V(G^b) &= V((G \setminus T) \times K_2) \cup T \\ E(G^b) &= E((G \setminus T) \times K_2) \cup \{\{t, t'\} \mid t, t' \in T \wedge \{t, t'\} \in E(G)\} \\ &\cup \ \{\{t, (v, x)\} \mid t \in T \wedge x \in V(K_2) \wedge v \in V(G) \wedge \{t, v\} \in E(G)\}, \end{aligned}$$

where $V(K_2) = \{1, 2\}$. It is easy to see that G^b contains a k-linkage that links A and B. Let G' be a minimal induced subgraph of G^b that contains a k-linkage L' that links A and B. From Theorem 5.1, it follows that

$$\mathbf{tw}(G') \leq w(k). \tag{5.1}$$

From now on we work towards proving that $G \leq_m G'$. In order to achieve this, we prove the following two claims for G'.

Claim 7. *If L' is a k-linkage in G' that links A and B then for every vertex $v \in V(G) \setminus T$ no path of L' contains both $(v, 1)$ and $(v, 2)$.*

Proof. Towards a contradiction, assume that for some vertex $v \in V(G) \setminus T$, there exists a (t, t')-path P of L' that contains both $(v, 1)$ and $(v, 2)$. Without loss of generality, assume also that $(v, 1)$ appears before $(v, 2)$ in P. Let y be the successor of $(v, 2)$ in P and notice that $y \neq (v, 1)$. From the definition of G^b and the fact that G' is an induced subgraph of G^b, $\{y, (v, 1)\} \in E(G') \setminus E(L')$. By replacing the subpath of P from $(v, 1)$ to y with the edge $\{(v, 1), y\}$, we obtain a linkage in $G' \setminus (v, 2)$ that links A and B. This contradicts to the minimality of G'. □

Claim 8. *If L' is a k-linkage in G' that links A and B then for every vertex $v \in V(G) \setminus T$, $V(L') \cap \{(v, 1), (v, 2)\} \neq \emptyset$.*

Proof. Assume, in contrary, that there exists a linkage L' in G' and a vertex $x \in V(G) \setminus T$ such that L' links A and B and $V(L') \cap \{(x,1),(x,2)\} = \emptyset$. Claim 7 ensures that, after contracting the edges $\{(v,1),(v,2)\}$, $v \in V(G) \setminus T$ (whenever they exist), the corresponding paths compose a 2-approximate k-linkage $\tilde{L}'$ of $G \setminus \{x\}$ that links A and B. This is a contradiction to the assumption that $\tilde{L}$ is unique. Thus, the claim holds. □

Recall that $T \subseteq V(G')$ and that G' is an induced subgraph of G^b. Claim 8 implies that we may obtain G from G' by contracting the edges $\{(v,1),(v,2)\}$ for every $v \in V(G) \setminus T$ (whenever they exist). As $G \leq_m G'$, from (5.1), it follows that, $\mathbf{tw}(G) \leq w(k)$. □

We remark that, the previous lemma holds for any graph G that contains an r-approximate k-linkage. This can be seen by substituting $(G \setminus T) \times K_2$ with $(G \setminus T) \times K_r$ in the proof.

We now state a lemma that provides an upper bound on the tree-width of a graph G, given an upper bound on the tree-width of its line graph $L(G)$.

Lemma 5.6. *Let k be a positive integer. If G is a graph with $\mathbf{tw}(L(G)) \leq k$ then $\mathbf{tw}(G) \leq 2k+1$.*

Proof. Suppose that G is graph such that $L(G)$ admits a tree decomposition of width at most k and recall that every vertex of $L(G)$ corresponds to an edge of G. We construct a tree decomposition $\mathcal{T}$ of G from a tree decomposition $\mathcal{T}_L$ of $L(G)$ by replacing in each bag of $\mathcal{T}_L$ every vertex of $L(G)$ by the endpoints of the corresponding edge in G. It is easy to verify that this is a tree decomposition of G. Therefore, $\mathbf{tw}(G) \leq 2k+1$. □

Before we proceed to the next lemma, we need to introduce the notion of an r-approximate k-edge-linkage in a graph. Similarly to the notion of an r-approximate linkage, an *r-approximate edge-linkage* in a graph G is a family of paths E in G such that for every $r+1$ distinct paths $P_1, P_2, \ldots, P_{r+1}$ in E, it holds that $\cap_{i\in[r+1]} E(P_i) = \emptyset$. We call these paths the *components* of the edge-linkage. Let $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $(\beta_1, \beta_2, \ldots, \beta_k)$ be elements of $V(G)^k$. We say that an r-approximate edge-linkage E, consisting of the paths $P_1, P_2, \ldots, P_k$, *links* $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $(\beta_1, \beta_2, \ldots, \beta_k)$ if P_i is a path with endpoints α_i and β_i, for every $i \in [k]$. The *order* of E is k. We call an r-approximate edge-linkage of order k, *r-approximate k-edge-linkage.* When $r = 1$, we call such a family of paths, an *edge-linkage.*

Lemma 5.7. *There exists a computable function r such that the following holds. Let G_1, G_2 and G be graphs such that $G_i \leq_{im} G$, $i \in [2]$. If G' is a minimal subgraph of G where $G_i \leq_{im} G'$, $i \in [2]$, then $\mathbf{tw}(G') \leq r(|E(G_1)|, |E(G_2)|)$.*

Proof. Let G' be a minimal subgraph of G such that $G_i \leq_{im} G'$, $i \in [2]$. Notice that the edges of G_i compose a k_i-edge-linkage E_i in G, where $k_i = |E(G_i)|$, $i \in [2]$. Furthermore, observe that the paths of E_1 and E_2 constitute a 2-approximate k-edge-linkage E of G, where $k = k_1 + k_2$. Indeed, notice that in contrary to linkages, we do not require the paths that are forming edge-linkages to have different endpoints. The minimality of G' implies that $\bigcup\{P \mid P \in E\} = G'$. Denote by $A = (v_{i_1}, v_{i_2}, \ldots, v_{i_k})$ and $B = (v_{j_1}, v_{j_2}, \ldots, v_{j_k})$ the vertex sets that are edge-linked by E in G' and let $\widehat{G}$ be the graph with

$$
\begin{aligned}
V(\widehat{G}) &= V(G') \cup \{u_{i_q} \mid q \in [k]\} \cup \{u_{j_q} \mid q \in [k]\}, \\
E(\widehat{G}) &= E(G') \cup \{t_{i_q} \mid q \in [k]\} \cup \{t_{j_q} \mid q \in [k]\},
\end{aligned}
$$

where the vertices u_{i_q} and u_{j_q}, $q \in [k]$, are new, $t_{i_q} = \{u_{i_q}, v_{i_q}\}$, $q \in [k]$, and $t_{j_q} = \{u_{j_q}, v_{j_q}\}$, $q \in [k]$.

Consider the line graph of $\widehat{G}$, $L(\widehat{G})$, and notice that E corresponds to a 2-approximate k-linkage L from A_L to B_L in $L(\widehat{G})$, where $A_L = (t_{i_1}, t_{i_2}, \ldots, t_{i_k})$ and $B_L = (t_{j_1}, t_{j_2}, \ldots, t_{j_k})$. This is true as, from the construction of $\widehat{G}$, all the vertices in A_L and B_L are distinct. The minimality of G' yields that $V(L) = V(L(\widehat{G}))$ and implies that L is unique. From Lemma 5.5, we obtain that $\mathbf{tw}(L(\widehat{G})) \leq f(k)$. Therefore, from Lemma 5.6, we get that $\mathbf{tw}(\widehat{G}) \leq p(f(k))$, where p is the function of Lemma 5.6. Finally, as $G' \subseteq \widehat{G}$, $\mathbf{tw}(G') \leq r(k_1, k_2)$, where $r(k_1, k_2) = p(f(k_1 + k_2))$. $\square$

Notice that Lemma 5.2 follows from Lemma 5.7 when we set G_2 to be the empty graph. Finally, we show that given two immersion-closed graph classes $\mathcal{C}_1$ and $\mathcal{C}_2$, the immersion-closed graph class $\mathcal{C}_1 \cup \mathcal{C}_2$ is layer-wise MSO-definable.

Observation 5.2. *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are immersion-closed graph classes then $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ is a layer-wise MSO-definable graph class defined, for every $k \geq 0$, by the formula*

$$\phi_k \equiv \left(\left(\bigwedge_{G \in \mathbf{obs}_{\leq_{im}}(\mathcal{C}_1)} \neg\phi_G \right) \vee \left(\bigwedge_{H \in \mathbf{obs}_{\leq_{im}}(\mathcal{C}_2)} \neg\phi_H \right) \right) \wedge \phi_{\mathcal{T}_k}$$

where ϕ_G and ϕ_H are the formulas described in Lemma 5.1, and $\phi_{\mathcal{T}_k}$ the formula of Theorem 5.3 .

We are now able to prove our main result.

Theorem 5.4. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two immersion-closed graph classes. If the sets $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1)$ and $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_2)$ are given then the set $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1 \cup \mathcal{C}_2)$ is computable.*

Proof. Observation 5.2 provides us with an MSO-description of the immersion-closed graph class $\mathcal{C}_1 \cup \mathcal{C}_2$, and Lemma 5.7 gives us an upper bound on the width of $\mathcal{C}_1 \cup \mathcal{C}_2$. Therefore, Lemma 5.4 is applicable. □

5.4 Conclusions

In this chapter, we further the study on the constructibility of obstruction sets for immersion-closed graph classes. In particular, we provide an upper bound on the tree-width of the obstructions of a graph class $\mathcal{C}$, which is the union of two immersion-closed graph classes $\mathcal{C}_1$ and $\mathcal{C}_2$ with $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_1)$ and $\mathbf{obs}_{\leq_{im}}(\mathcal{C}_2)$ given. Then, using that result, we prove that $\mathbf{obs}_{\leq_{im}}(\mathcal{C})$ is computable.

In [211], N. Robertson and P. Seymour claimed that the class of graphs is also well-quasi-ordered under the strong immersion ordering. However, a full proof of this result has not appeared so far. We remark that the combinatorial results of this chapter, that is, the upper bounds on the tree-width of the obstructions, also hold for the strong immersion ordering. Thus, if the claim of N. Robertson and P. Seymour holds, the obstruction set of the union of two strongly immersion-closed graph classes, whose obstruction sets are given, can be effectively computed.

Finally, it was proven by B. Courcelle, R. Downey and M. Fellows [37] that the obstruction set of a minor-closed graph class $\mathcal{C}$ cannot be computed by an algorithm whose input is a description of $\mathcal{C}$ as an MSO-sentence. The computability of the obstruction set of an immersion-closed graph class $\mathcal{C}$, given solely an MSO description of $\mathcal{C}$, remains an open problem.

CHAPTER 6

The Graph Minors Weak Structure Theorem

The Graph Minors series of Robertson and Seymour appeared to be a rich source of structural results in Graph Theory with multiple applications in Algorithms. One of the most celebrated outcomes of this project was the existence of an $O(n^3)$ step algorithm for solving problems such as the Disjoint Path and the Minor Containment. A basic ingredient of these algorithms is a theorem, proved in paper XIII of the series [193], revealing the local structure of graphs excluding some graph as a minor. This result, now called the *weak structure theorem*, asserts that there is a function $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that for every integer k, every h-vertex graph H, and every graph G, one of the following holds:

1. G contains H as a minor,

2. G has treewidth at most $f(k, h)$, or

3. G contains a set X of at most $\binom{h}{2}$ vertices (called *apices*) such that $G \setminus X$ contains as a subgraph the subdivision W of a wall of height k that is arranged inside G in a "flat" manner (*flatness condition*).

To make the above statement precise we need to clarify the flatness condition in the third statement above. We postpone this complicated task until Section 6.2 and instead, we roughly visualize W in a way that the part of $G \setminus X$ that is located inside the perimeter P of W can be seen as a set of graphs attached on a plane region where each of these graphs has bounded treewidth and its boundary with the other graphs is bounded by 3.

The algorithmic applications of the weak structure theorem reside in the fact that the graph inside P can be seen as a *bidimensional structure* where, for several combinatorial problems, a solution certificate can be revised so that it avoids the middle vertex of the subdivided wall W. This is known as the *irrelevant vertex technique* and can be seen as a reduction of an instance of a problem to an equivalent one where this "irrelevant vertex" has been deleted. The application of this technique has now gone much further than its original use in the Graph Minors series and has evolved to a standard tool in Algorithmic Graph Minors Theory (see [42, 44, 106, 133, 134, 138] for applications of this technique).

In this chapter we prove an optimized version of the weak structure theorem. Our improvement is twofold: first, the function f is now linear on k and second, the number of the apices is bounded by $h-5$. Both our optimizations are optimal as indicated by the graph J obtained by taking a $(k \times k)$-grid (for $k \geq 3$) and making all its vertices adjacent with a copy of K_{h-5}. Indeed, it is easy to verify that J excludes $H = K_h$ as a minor, its treewidth is $k+h-5$ and becomes planar (here, this is equivalent to the "flatness" condition) after the removal of exactly $h-5$ vertices.

Our proof deviates significantly from the one in [193]. It builds on the (strong) *structure theorem* of the Graph Minors that was proven in paper XVI of the series [207]. This theorem reveals the global structure of a graph without a K_h as a minor and asserts that each such graph can be obtained by gluing together graphs that can "almost be embedded" in a surface where K_h cannot be embedded (see Section 6.1 for the exact statement). The proof exploits this structural result and combines it with the fact, proved in [83], that apex-free "almost embedded graphs" without a $(k \times k)$-grid have treewidth $O(k)$.

The organization of this chapter is the following. In Section 6.1 we give the definitions of all the tools that we are going to use in our proof, including the Graph Minors structure theorem. The definition of the flatness condition is given in Section 6.2, together with the statement of our main result. Some lemmata concerning the invariance of the flatness property under certain local transformations are given in Subsection 6.3.1 and further definitions and results concerning apex vertices are given in Subsection 6.3.2. The proof of our main result is presented in Section 6.4. Finally, in Section 6.5 we see how two already known results and a new one can be obtained from our main result and how the duality and self-duality of some regular tilings can be "expanded" to the realm of the graphs that exclude a fixed apex graph as a minor.

6.1 Preliminaries

Graph Minors structure theorem. The proof of our result is using the Excluded Minor Theorem from the Graph Minors. Before we state it, we need some definitions.

Definition 6.1 (*h-nearly embeddable graphs*)**.** Let Σ be a surface and

$h > 0$ be an integer. A graph G is *h-nearly embeddable* in Σ if there is a set of vertices $X \subseteq V(G)$ (called *apices*) of size at most h such that graph $G - X$ is the union of (possibly empty) subgraphs $G_0, \ldots, G_h$ with the following properties:

i) There is a set of cycles $C_1, \ldots, C_h$ in Σ such that the cycles C_i are the borders of open pairwise disjoint discs Δ_i in Σ;

ii) G_0 has an embedding in Σ in such a way that $G_0 \cap \bigcup_{i=1,\ldots,h} \Delta_i = \emptyset$;

iii) graphs $G_1, \ldots, G_h$ (called vortices) are pairwise disjoint and for $1 \leq i \leq h$, $V(G_0) \cap V(G_i) \subset C_i$;

iv) for $1 \leq i \leq h$, let $U_i := \{u_1^i, \ldots, u_{m_i}^i\}$ be the vertices of $V(G_0) \cap V(G_i) \subset C_i$ appearing in an order obtained by clockwise traversing of C_i. We call vertices of U_i *bases* of G_i. Then G_i has a path decomposition $\mathcal{B}_i = (B_j^i)_{1 \leq j \leq m_i}$, of width at most h such that for $1 \leq j \leq m_i$, we have $u_j^i \in B_j^i$.

Given a tree decomposition $(\mathcal{X}, T)$ of a graph G, where $\mathcal{X}$ is the set of the bags of the decomposition, for every $i \in V(T)$ we denote by $\overline{X}_i$ the *closure* of the bag $X_i \in \mathcal{X}$, that is, $\overline{X}_i$ is the graph $G\left[X_i\right] \cup \left(\cup_{j \in N_T(i)} K\left[X_i \cap X_j\right]\right)$. (We would like to mention here that the graph $\overline{X}_i$ is also referred as *torso* at node i.)

Observation 6.1. *If G is a graph and $(\mathcal{X}, T)$ is a tree decomposition of G then there exists an $X \in \mathcal{X}$ such that $\mathbf{tw}(\overline{X}) \geq \mathbf{tw}(G)$.*

We also need the simple following result.

Lemma 6.1. *If G is a graph and $X \subseteq V(G)$, then $\mathbf{tw}(G - X) \geq \mathbf{tw}(G) - |X|$.*

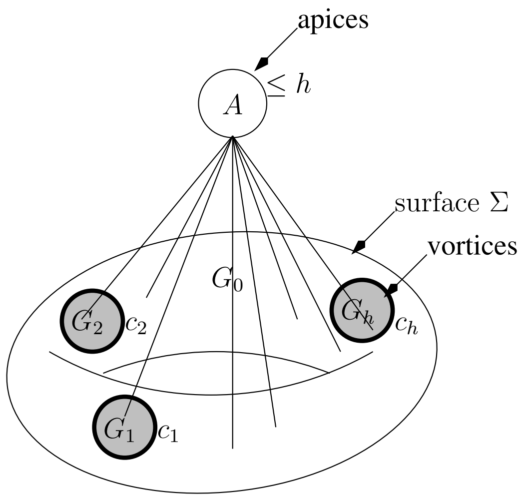


Figure 6.1: An h-nearly embeddable graph.

We say that a tree decomposition $(\mathcal{X}, T)$ of a graph G is *small* if for every $i, j \in V(T)$, with $i \neq j$, $X_i \not\subseteq X_j$.

A simple proof of the following lemma can be found in [82].

Lemma 6.2. *The following hold.*

1. *If G is a graph and $(\mathcal{X}, T)$ is a small tree decomposition of G then $|V(T)| \leq |V(G)|$.*

2. *Every graph G has a small tree decomposition of width $\mathbf{tw}(G)$.*

The following proposition is known as the Graph Minors structure theorem [207]. (For an example[1] of an H-minor-free graph, see Figure 6.2)

[1] As the figures only aim to facilitate intuition the surface Σ is depicted as a torus. Note, however, that we work on general surfaces of bounded genus.

Proposition 6.1. *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that, for every non-planar graph H with h vertices and every graph G excluding H as a minor there exists a tree decomposition $(\mathcal{G} = \{G_i \mid i \in V(T)\}, T)$ where for every $i \in V(T)$, $\overline{G}_i$ is an $f(h)$-nearly embeddable graph in a surface Σ on which H cannot be embedded.*

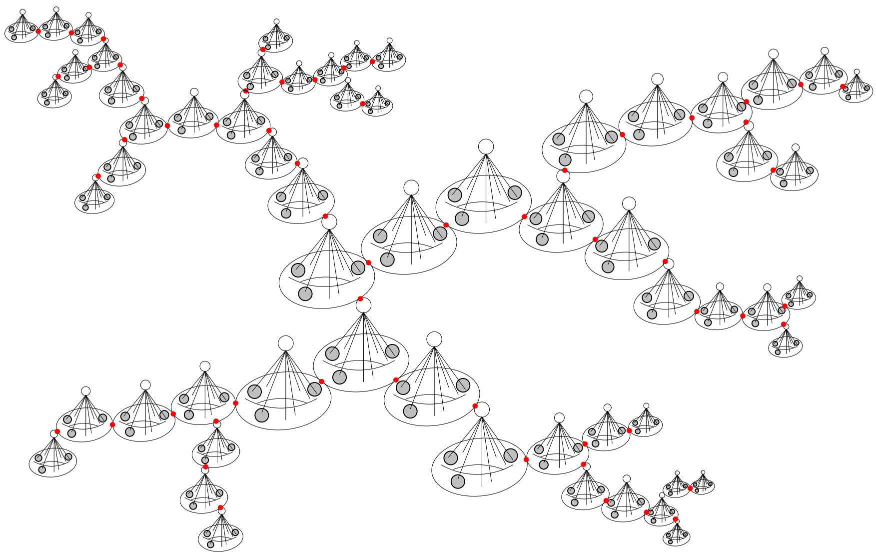


Figure 6.2: An example of an H-minor-free graph.

6.2 Statement of the Main Result

We define Γ_k as the following (unique, up to isomorphism) triangulation of the $(k \times k)$-grid. Let Γ be a plane embedding of the $(k \times k)$-grid such that all external vertices are on the boundary of the external face. We triangulate internal faces of the $(k \times k)$-grid such that, in the obtained graph, all the internal vertices have degree 6 and all non-corner external vertices have degree 4. The construction of Γ_k is completed if we connect one corner of degree two with all vertices of the external face (we call this corner *loaded*). For an example, see Γ_6 in Figure 6.3. We also use notation Γ_k^* for the graph obtained from Γ_k if we remove all edges incident to its loaded vertex that do not exist in its underlying grid.

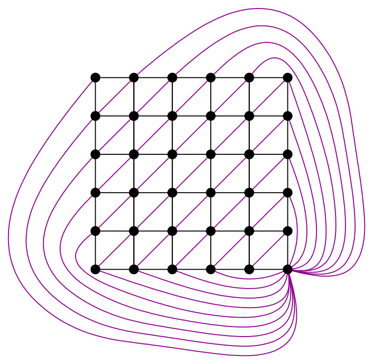


Figure 6.3: The graph Γ_6.

We define the (k, l)-*pyramid* to be the graph obtained if we take the disjoint union of a $(k \times k)$-grid and a clique K_l and then add all edges between the vertices of the clique and the vertices of the grid. We denote the (k, l)-pyramid by $\Pi_{k,l}$.

Compasses and rural devisions. Let W be a subdivided wall in a graph G. We say that W is *flat* in G if its compass K in G has no (c_1, c_3)-path and (c_2, c_4)-path that are vertex-disjoint.

Observation 6.2. *If G is a graph containing a flat wall W as a subgraph then any subdivision of W is also flat in the graph G' obtained from G by the subdivisions of the wall.*

If J is a subgraph of K, we denote by $\partial_K J$ the set of all vertices v such that either $v \in C$ or v is incident with an edge of K that is not in J, that is,

$$\partial_K J = \{v \in V(J) \mid v \in C \text{ or } \exists e \in E(K) \setminus E(J) : v \in e\}.$$

A *rural division* $\mathcal{D}$ of the compass K is a collection

$$(D_1, D_2, \ldots, D_m)$$

of subgraphs of K with the following properties:

1. $\{E(D_1), E(D_2), \ldots, E(D_m)\}$ is a partition of $E(K)$ into non-empty subsets,

2. For $i, j \in [m]$, if $i \neq j$ then $\partial_K D_i \neq \partial_K D_j$ and $V(D_i) \cap V(D_j) = \partial_K D_i \cap \partial_K D_j$,

3. For each $i \in [m]$ and all $x, y \in \partial_K D_i$ there exists a (x, y)-path in D_i with no internal vertex in $\partial_K D_i$,

4. For each $i \in [m]$, $|\partial_K D_i| \leq 3$, and

5. The hypergraph $H_K = (\bigcup_{i \in [m]} \partial_K D_i, \{\partial_K D_i \mid i \in [m]\})$ is planar, its incidence graph can be embedded in a closed disk Δ such that c_1, c_2, c_3, and c_4 appear in this order on the boundary of Δ and for each hyperedge e of H there exist $|e|$ mutually vertex-disjoint paths between e and C in K.

We call the elements of $\mathcal{D}$ *flaps*. A flap $D \in \mathcal{D}$ is *internal* if $V(D) \cap V(P) = \emptyset$.

We can now state the main result of this chapter.

Theorem 6.1. *There exists a computable function f such that, for every two graphs H and G and every $k \in \mathbb{N}$, one of the following holds:*

1. *H is a minor of G.*

2. *$\mathbf{tw}(G) \leq f(h) \cdot k$, where $h = \mathbf{n}(H)$.*

3. *$\exists A \subseteq V(G)$ with $|A| \leq \mathbf{an}(H) - 1$ such that $G \setminus A$ contains as a subgraph a flat subdivided wall W where*

   - *W has height k and*
   - *the compass of W has a rural division $\mathcal{D}$ such that each internal flap of $\mathcal{D}$ has treewidth at most $f(h) \cdot k$.*

We postpone the proof of Theorem 6.1 until Section 6.4 and we conclude this section with a brief description of the proof. A main ingredient is the Strong Structural Theorem, that is, Proposition 6.1, asserting that every H-minor free graph G can be seen as a tree decomposition such that, for every node G_i, the graph $\overline{G_i}$ is a $f(h)$-nearly embeddable graph. Given that the graph G has treewidth at least $f(h) \cdot k$ where $f(h)$ is "big enough" (depending on the excluded graph H), there should exist a node G_i of the tree decomposition such that the treewidth of $\overline{G_i}$ is still big enough while all other nodes have smaller treewidth. This, according to the result of [83], implies that the $f(h)$-nearly embeddable graph $\overline{G_i}$ contains as a subgraph the subdivision W of a "big enough" (but still depending linearly on k) wall that is flatly embedded in a surface, in the sense that its perimeter is the boundary of a disk whose interior, the compass of W, contains, among other parts of $\overline{G_i}$, the rest of W.

Our next, and more technical, step is to extract from this "local" structure, concerning $\overline{G_i}$, a wall and a rural division of its compass in the general graph G. For this, we treat all other parts of the tree decomposition as flaps of bounded treewidth that contain at most three non-apex vertices of $\overline{G}_i$.

At this point, it follows that the third assertion of Theorem 6.1 holds for the augmented graph G' that is obtained from G if we add all "virtual" edges that are edges of $\overline{G}_i$ but not of G_i. The proof concludes by showing that, even if the removal of these virtual edges my harm parts of the subdivided wall W and the corresponding rural division, these parts can be reconstructed by collections of paths inside the flaps that are now attached on the compass of $\overline{G}_i$.

6.3 Some Auxiliary Lemmata

The main results in this section are Lemmata 6.6 and 6.10 that are used for the proof of our main result in Section 6.4.

6.3.1 An Invariance Lemma for Flatness

Before proving the main results of this section we need to state the following folklore result. (See, for example, Proposition 1.7.2 in [60].)

Proposition 6.2. *Let G and H be graphs such that $H \leq_m G$. If $\Delta(H) \leq 3$, then $H \leq_{tm} G$.*

We also need the following definition. Let $G = G_0 \cup G^+$, where G_0 is a graph embedded in a surface Σ of Euler genus γ and let G^+ is another graph that might share common vertices with G_0. Let also H be a graph and $v \in V(H)$. We say that G *contains H as a v-smooth contraction* if $H \leq_c^{\phi} G$ for some $\phi : V(G) \to V(H)$ and there exists a closed disk D in Σ such that the vertices of G^+ are outside of D and all the vertices of G that are outside D are exactly the model of v, that is, $\phi^{-1}(v) = V(G) \setminus (V(G) \cap D)$.

Lemma 6.3. *Let k be a positive integer and G be a graph that is h-nearly embedded in a surface of Euler genus γ without apices and contains $\Gamma_{2\cdot k+8}$ as a v-smooth contraction, where v is the loaded corner of $\Gamma_{2\cdot k+8}$. Then G contains as a subgraph a subdivided wall of height k whose compass can be embedded in a closed disk Δ such that the perimeter of W is identical to the boundary of Δ.*

Proof. Assume that $\Gamma_{2\cdot k+8}$ is a v-smooth contraction of G via ϕ, where v is the loaded corner of $\Gamma_{2\cdot k+8}$. Without loss of generality, let

$$V(\Gamma_{2\cdot k+8}) = \{1, \ldots, 2\cdot k+8\}^2,$$

where $v = (2\cdot k+8, 2\cdot k+8)$. Let R be the set of external vertices of $\Gamma_{2\cdot k+8}$ and let $G' = G \setminus \bigcup_{x\in R} \phi^{-1}(x)$. As G contains $\Gamma_{2\cdot k+8}$ as a v-smooth contraction and $v \in R$, it follows that G' is embedded inside an open disk Δ'. Moreover G' can be contracted to $\Gamma^*_{2\cdot k+6}$ via the restriction of ϕ to $V(G')$. From the definition of a wall, it follows that $\Gamma^*_{2\cdot k+6}$ contains W_{k+2} as a subgraph. As G' contains $\Gamma^*_{2\cdot k+6}$ as a minor, it follows that G' contains W_{k+2} as a minor. Evenmore, as W_{k+2} has maximum degree 3, from Proposition 6.2, it is also a topological minor of G'. Therefore G' contains as a subgraph (embedded in Δ') a subdivided wall of height $k+2$. Among all such subdivided walls, let W_{ex} be the one whose compass has the minimum number of faces inside the annulus $\Phi = \Delta_{\text{ex}} \setminus \Delta \subseteq \Delta'$ where Δ_{ex} and Δ are defined as the closed disks defined so that the boundary of Δ_{ex} is the first layer of W_{ex} and the boundary of Δ is the second one.

Let W be the subdivided wall of G' whose perimeter is the boundary of Δ. By definition, all vertices of the compass K of W are inside Δ. It now remains to prove that the same holds also for the edges of K. Suppose in contrary that $\{x, y\}$ is an edge outside Δ. Clearly, both x and y lie on the boundary of Δ and $\{x, y\}$ is inside the disk Δ^* defined by some brick of W_{ex} that is inside Φ. We distinguish two cases:

Case 1: $\{x, y\}$ are in the same brick, say A of W. Then, there is a path of this brick that can be replaced in W by $\{x, y\}$ and substitute W by a new subdivided wall corresponding to an annulus with less faces, a contradiction. (See, Figure 6.4.)

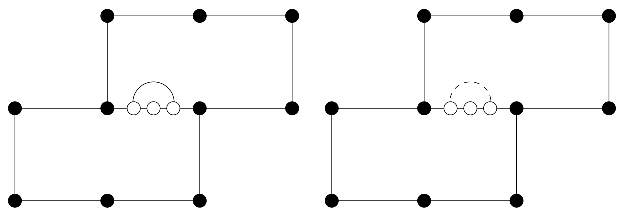


Figure 6.4: Example of Case 1 in Lemma 6.3.

Case 2: $\{x, y\}$ are not in the same brick of W. Then x and y should belong in neighboring bricks, say B and C respectively. Let A be the unique brick of W_{ex} that contains x and y and w be the unique common vertex in A, B and C. Observe that there a path P_B of B connecting x and w and a path P_C of C connecting y and w. Then we substitute W by a new wall as follows: we replace w by x, P_C by $\{x, y\}$, and see P_B as a subpath of the common path between B and C. (See, Figure 6.5.)

Again, the new wall corresponds to an annulus with less faces, a contradiction. □

Lemma 6.4 ([83]). *There is a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that if G is a graph h-nearly embedded in a surface of Euler genus γ without apices,*

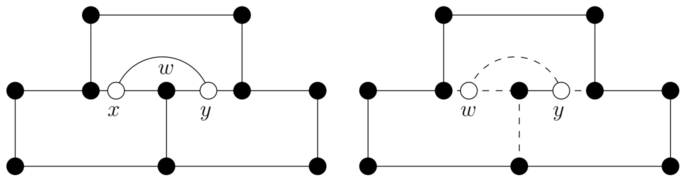


Figure 6.5: Example of Case 2 in Lemma 6.3.

where $\mathbf{tw}(G) \geq f(\gamma, h) \cdot k$*, then* G *contains as a* v*-smooth contraction the graph* Γ_k *with the loaded corner* v*.*

Lemma 6.5. *Let* h *be a positive integer and* G *be a graph that contains a flat subdivided wall* W *of height* h*. If* K_3 *is a subgraph of the compass of* W *then after applying a* ΔY*-transformation in* K_3 *the resulting graph also contains a flat subdivided wall* W' *of height* h *as a subgraph. Moreover,* W' *is isomorphic to a subdivision of* W*.*

Proof. We examine the non-trivial case where $E(K_3) \cap E(W) \neq \emptyset$. Observe that, as W does not contain triangles, $|E(K_3) \cap E(W)| < 3$. In what follows we denote by x, y, z the vertices of K_3, w the vertex that appears after the transformation, and distinguish the following cases.

Case 1. K_3 and W have exactly one common edge, say $\{x, y\}$. As w is a new vertex, the path (x, w, y) that appears after the ΔY-transformation has no common internal vertices with W. In this case, we replace the edge $\{x, y\}$ in W by the edges $\{x, w\}, \{w, y\}$.

Case 2. K_3 and W have exactly two common edges, say $\{x, y\}$ and $\{x, z\}$. We distinguish the following two subcases.

Subcase 2.1. x is an original vertex and x is not a corner of W. Let q be the third vertex in the neighborhood of x. Observe that the ΔY-transformation is equivalent to removing the edge $\{y, z\}$, which is not an edge of the wall, and subdividing the edge $\{x, q\}$. Then the lemma follows from Observation 6.2. (See, Figure 6.6)

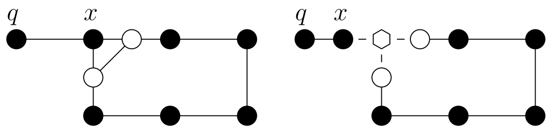


Figure 6.6: Example of Subase 2.1 in Lemma 6.5.

Subcase 2.2. x is not an original vertex or x is a corner. Then the ΔY-transformation is equivalent to removing the edge $\{y, z\}$, which is not an edge of W, and then substituting $\{y, x\}$ by $\{y, w\}$ and $\{x, z\}$ by $\{w, z\}$. (See, Figure 6.7)

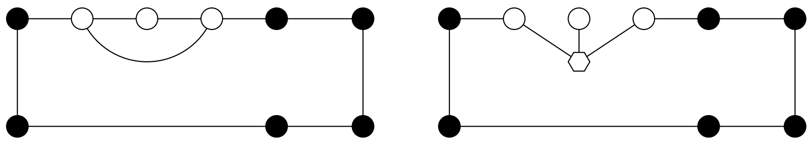


Figure 6.7: Example of Subcase 2.2 in Lemma 6.5.

Observe that in all the above cases the resulting wall W' remains flat and is isomorphic to a subdivision of W and the lemma follows. □

By applying inductively Observation 6.2 and Lemma 6.5 we derive the following.

Lemma 6.6. *Let h be a positive integer and G be a graph that contains a flat subdivided wall W of height h as a subgraph. If we apply a sequence of subdivisions or ΔY-transformations in G, then the resulting graph will contain a flat subdivided wall W' of height h as a subgraph. Moreover, W' is isomorphic to a subdivision of W.*

6.3.2 Pyramids and Tree-width

Let us first state the next.

Proposition 6.3 ([197]). *Let n be a positive integer. If H is a planar graph with $|V(H)|+2|E(H)| \leq n$, then H is isomorphic to a minor of the $2n \times 2n$-grid.*

Combining Proposition 6.3 with Eűler's formula for planar graphs we obtain the following.

Lemma 6.7. *If G is a planar graph then G is isomorphic to a minor of the $(14 \cdot \mathbf{n}(G) - 24) \times (14 \cdot \mathbf{n}(G) - 24)$-grid.*

From the above lemma we obtain the following.

Lemma 6.8. *Let h be an integer. If G is an h-apex graph then G is isomorphic to a minor of $\Pi_{14\cdot(\mathbf{n}(G)-h)-24,h}$.*

Lemma 6.9. *Let G be the graph obtained by a $((k+\lceil\sqrt{h}\rceil)\times(k+\lceil\sqrt{h}\rceil))$-grid if we make its vertices adjacent to a set A of h new vertices. Then G contains $\Pi_{k,h}$ as a minor.*

Proof. We denote by G' the grid used for constructing G and let G_1 and G_2 two disjoint subgraphs of G' where G_1 is isomorphic to a $(k \times k)$-grid and G_2 is isomorphic to a $(\alpha \times \alpha)$-grid where $\alpha = \lceil\sqrt{h}\rceil$. Remove from G all vertices not in $A \cup V(G_1) \cup V(G_2)$. Then remove all edges of G' incident to $V(G_2)$ and notice that in the remaining graph F, the vertices in $A \cup V(G_2)$ induce a graph isomorphic to K_{h,α^2} which, in turn, can be contracted to a clique on the vertices of A. Applying the same contractions in F one may obtain $\Pi_{k,h}$ as a minor of G. $\square$

Lemma 6.10. *Let G, H be graphs such that H is not a minor of G and there exists a set $A \subseteq V(G)$ such that $G \setminus A$ contains a wall W of height $g(h)\cdot(k+1)-1$ as a subgraph, where $g(h) = 14\cdot(h-\mathbf{an}(H))+\lceil\sqrt{\mathbf{an}(H)}\rceil - 24$ and $h = \mathbf{n}(H)$. If $|A| \geq \mathbf{an}(H)$ then there exists an $A' \subseteq A$ such that $G \setminus A'$ contains a wall $W' \subseteq W$ of height k as a subgraph with the property that if K' is the compass of W' in $G \setminus A'$ then $V(K') \cap (A \setminus A') = \emptyset$. Evenmore, $|A'| < |A|$.*

Proof. Let $A = \{\alpha_i \mid i \in [|A|]\}$ and $P_{g(h)} = \{W_{(m,l)} \mid (m,l) \in [g(h)]^2\}$ be a collection of $(g(h))^2$ disjoint subwalls $W_{(m,l)}, (m,l) \in [g(h)]^2$ of W with height k. For every $(m,l) \in [g(h)]^2$, we denote by $K_{(m,l)}$ the compass of $W_{(m,l)}$ in $G \setminus A$ and let $q_{(m,l)} = (q^1_{(m,l)}, q^2_{(m,l)}, \ldots, q^{|A|}_{(m,l)})$ be the binary vector where for every $j \in |A|$,

$$q^j_{(m,l)} = \begin{cases} 1 & \text{if } \exists v \in V(K_{(m,l)}) : \{v, \alpha_j\} \in E(G) \\ 0 & \text{if } \forall v \in V(K_{(m,l)}) : \{v, \alpha_j\} \notin E(G) \end{cases}$$

We claim that there exists an $(m', l') \in [g(h)]^2$ such that $q_{(m',l')} \neq (1, 1, \ldots, 1)$. Indeed, assume in contrary, that for every $(m,l) \in [g(h)]^2$, $q_{(m,l)} = (1, 1, \ldots, 1)$. We will arrive to a contradiction by showing that H is a minor of G. For this, consider the graph

$$G' = G[V(W) \cup \bigcup_{(m,l)\in[g(h)]^2} V(K_{(m,l)})] \subseteq G.$$

For every $(m,l) \in [g(h)]^2$, contract each $K_{(m,l)}$ to a single vertex and this implies the existence of a $(g(h) \times g(h))$-grid as a minor of G' and therefore of $G \setminus A$ as well. Moreover, for each vertex v of this grid it holds that each vertex in A is adjacent to some vertex of the model of v, therefore G contains the graph J obtained after we take a $(g(h) \times g(h))$-grid and connect all its vertices with $\mathbf{an}(H)$ new vertices. From Lemma 6.9, G

contains $\Pi_{14\cdot(n(h)-\mathbf{an}(H))-24,\mathbf{an}(H)}$ as minor. Applying now Lemma 6.8, we obtain that G contains H as a minor, a contradiction. Therefore, there exist $(m',l') \in [g(h)]^2$ and $j_0 \in [|A|]$ such that $q^{j_0}_{(m',l')} = 0$. The lemma follows for $A' = A \setminus \{\alpha_{j_0}\}$ and $W' = W_{(m',l')}$. $\square$

6.4 The Main Proof

6.4.1 Notation

Below, we define the notation which is useful to the proof of the main result.

Given a tree decomposition $\mathcal{T} = (\mathcal{X} = \{X_i \mid i \in V(T)\}, T)$ of a graph G a vertex $i_0 \in V(T)$ and a set of vertices $I \subseteq N_T(i_0)$, we define $\mathcal{T}_{i_0,I}$ as the collection of connected components of $T \setminus i_0$ that contain vertices of I. Given a subtree Y of T, we define $G_Y = G[\cup_{i\in V(Y)} X_i]$ and $\overline{G}_Y = \cup_{i\in V(Y)} \overline{X}_i$.

Observation 6.3. *Given a tree decomposition $\mathcal{T} = (\mathcal{X} = \{X_i \mid i \in V(T)\}, T)$ of a graph G, a vertex $i_0 \in V(T)$, and a set of vertices $I \subseteq N_T(i_0)$, it holds that for every $T_1, T_2 \in \mathcal{T}_{i_0,I}$, $\overline{G}_{T_1} \cap \overline{G}_{T_2}$ is a complete graph.*

Given a family of graphs $\mathcal{F}$, a graph G and a set of vertices $S \subseteq V(G)$, we define the class $\mathcal{F}^*_{S,G}$ as the collection of the connected components in the graphs of $\mathcal{F} \setminus S$ and the class $\mathcal{F}_{S,G}$ as the set of graphs in $\mathcal{F}^*_{S,G}$ that have some common vertex with $G \setminus S$. We say that two graphs $G_1, G_2 \in \mathcal{F}_{S,G}$ are G-equivalent if $V(G_1) \cap V(G \setminus S) = V(G_2) \cap V(G \setminus S)$ and let $\mathcal{F}^1_{S,G}, \ldots, \mathcal{F}^\rho_{S,G}$ be the equivalence classes defined that way. We denote by $\mathcal{P}_{\mathcal{F},S,G} = \{\bigcup \mathcal{F}^1_{S,G}, \ldots, \bigcup \mathcal{F}^\rho_{S,G}\}$, that is, for each equivalence class $\mathcal{F}^i_{S,G}$ we construct a graph in $\mathcal{P}_{\mathcal{F},S,G}$, by taking the union of the graphs in $\mathcal{F}^i_{S,G}$.

6.4.2 Proof of the Main Result

Before proving the main result let us prove the following.

Observation 6.4. *Let T be a tree, $k \in \mathbb{N}$ and $w : V(T) \to \mathbb{N}$ such that there exists at least one vertex $v \in V(T)$ with $w(v) \geq k$. There exists a vertex $u \in V(T)$ with $w(u) \geq k$ such that at most one of the connected components of $T \setminus u$ contains a vertex u' with $w(u') > k$.*

Proof. Let $Y = \{v \in V(T) \mid w(v) \geq k\}$. Pick a vertex r of T and let v be a vertex of Y with maximum distance away from r. It is easy to verify that the lemma holds for v. □

We may now proceed with the main proof of this section.

Proof of Theorem 6.1. Let G be a graph that excludes H as a minor. By Proposition 6.1, there is a computable function f_1 such that there exists a tree decomposition

$$\mathcal{T} = (\mathcal{X} = \{X_i \mid i \in V(T)\}, T)$$

of G, where for every $i \in V(T)$, the graphs $\overline{X}_i$ are $f_1(h)$-nearly-embeddable in a surface Σ of genus $f_1(h)$. Among all such tree-decompositions we choose $\mathcal{T} = (\mathcal{X}, T)$ such that:

(i) $\mathcal{T}$ is small.

(ii) Subject to (i), $\mathcal{T}$ has maximum number of nodes.

(iii) Subject to (ii), the quantity $\sum_{\substack{i,j \in V(T)\\ i \neq j}} |X_i \cap X_j|$ is minimized.

Notice that, from Lemma 6.2, Condition (i) guaranties the possibility of the choice of Condition (ii). We use the notation $\overline{G}$ to denote the graph $\overline{G}_T$ and we call the edges of $E(\overline{G}) \setminus E(G)$ *virtual*.

Let $w : V(T) \to \mathbb{N}$ such that $w(i) = \mathbf{tw}(\overline{X}_i)$. Observation 6.4 and Observation 6.1 imply that there exists a vertex $i_0 \in V(T)$ such that $\mathbf{tw}(\overline{X}_{i_0}) \geq \mathbf{tw}(G)$ and at most one of the connected components of $T \setminus i_0$ contains a vertex j such that $w(j) > w(i_0)$. We denote by A_{i_0} the set of apices of the graph $\overline{X}_{i_0}$ and by F the graph $\overline{X}_{i_0} \setminus A_{i_0}$ (notice that $F \subseteq \overline{G}$ but F is not necessarily a subgraph of G as F may contain virtual edges).

From Lemma 6.1 and the choice of i_0 it holds that

$$\mathbf{tw}(F) = \mathbf{tw}(\overline{X}_{i_0} \setminus A_{i_0}) \geq \mathbf{tw}(\overline{X}_{i_0}) - |A_{i_0}| \geq \mathbf{tw}(G) - |A_{i_0}|. \tag{6.1}$$

Recall that $|A_{i_0}| \leq f_1(h)$. Let f_2 be the two-variable function of Lemma 6.4. We define the two-variable function f_3 and the one-variable functions f_4 and f_5 such that

$$\begin{aligned} f_5(h) &= 14 \cdot (h - \mathbf{an}(H)) + \lceil \sqrt{\mathbf{an}(H)} \rceil - 24 \\ f_4(h) &= f_5(h)^{|A_{i_0}| - \mathbf{an}(H) + 1} \\ f_3(h, k) &= f_2(f_1(h), f_1(h)) \cdot (4k \cdot f_4(h) + 12) + f_1(h) \end{aligned}$$

As F is $f_1(h)$-nearly embeddable in Σ and does not contain any apices, from (6.1) and Lemma 6.4, we obtain that if $\mathbf{tw}(G) \geq f_3(h, k)$ then F contains the graph $Q = \Gamma_{4k \cdot f_4(h) + 12}$ as a v-smooth contraction, where v is the loaded corner of Q. (See Figure 6.8.)

From Lemma 6.3, it follows that F contains as a subgraph a flat subdivided wall W' of height $2k \cdot f_4(h) + 2$ whose compass K' in F can be embedded in a closed disk Δ such that the perimeter of W' is identical to the boundary of Δ. Furthermore, notice here, that W' is inside the same

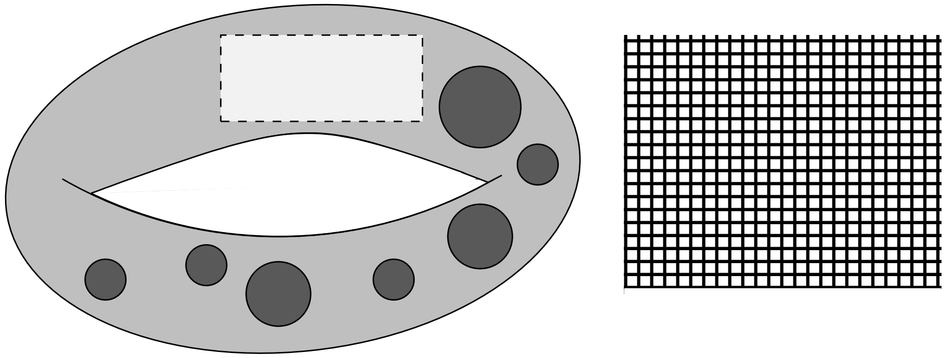


Figure 6.8: On the left: The graph F nearly embedded on Σ (without apices), where the dashed parallelogram represents the disk of the v-smooth contraction. On the right: The content of the disk.

disk where the preimage of the vertices of the v-smooth contraction where embedded. (See Figure 6.9.)

Let

$$I' = \{i \in N_T(i_0) \mid X_i \cap V(K') \neq \emptyset\}.$$

In other words, I' corresponds to all nodes of the tree decomposition $\mathcal{T}$, adjacent to i_0, that have vertices "inside" the compass of the subdivided wall W' in F (dashed parallelogram in Figure 6.9).

Claim 9. *For every $i \in I', |V(K') \cap X_i| \leq 3$.*

Proof of Claim 9. As K' is clearly planar, every clique in K' has size at most 4. Furthermore, K' does not contain K_4 as a subgraph.

Indeed, if so, one of the triangles of the clique would be a separator of G. We denote the vertices of this triangle by z_1, z_2, and z_3. Let then z be the vertex of the clique that is different from z_1, z_2, and z_3, and S be

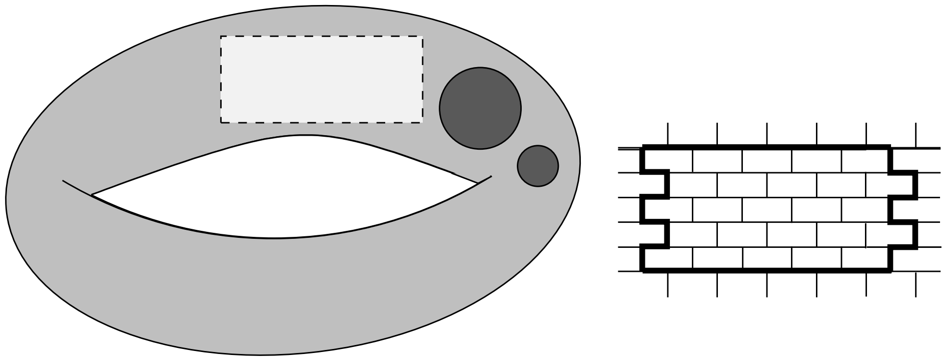


Figure 6.9: On the left: The graph F nearly embedded on Σ (without apices), where the dashed parallelogram represents the perimeter of the wall. On the right: Part the content of the disk.

the vertices contained in the interior of the disk defined by the separating triangle, where $z \in S$.

We may assume that there exists a vertex $j \in I'$ that contains all the vertices of the clique, that is, $\{z, z_1, z_2, z_3\} \subseteq X_j$. Indeed, towards a contradiction assume that there is no such vertex. Then, we may modify $\mathcal{T}$ in order to construct a small tree decomposition $\mathcal{T}'$ with more bags than $\mathcal{T}$ in the following way.

We add a new vertex j_0 in T, and a new bag X_{j_0} containing all vertices of the clique, the vertices belonging to the interior of the separating triangle of the clique, and all apices of X_{i_0}, that is $X_{j_0} = \{z_1, z_2, z_3\} \cup S \cup A_{i_0}$. We then remove S from X_{i_0}. Finally we add the edge $\{i_0, j_0\}$, remove all the edges between the neighbors $i \in I'$ of i_0 in T, whose common vertices with X_{i_0} are either apices or vertices of S and the clique, and make them adjacent to j_0 instead. Formally, let $\mathcal{T}' = (\mathcal{Y}, T')$, with

$$\begin{aligned}
V(T') &= V(T) \cup \{j_0\}, \text{ where } j_0 \notin V(T), \\
E(T') &= (E(T) \setminus \{\{i, i_0\} \mid i \in N_T(i_0) \\
&\qquad\qquad \wedge (X_i \cap X_{i_0}) \subseteq S \cup \{z_1, z_2, z_3\} \cup A_{i_0}\}) \cup \\
&\quad \{\{i, j_0\} \mid i \in N_T(i_0) \\
&\qquad\qquad \wedge (X_i \cap X_{i_0}) \subseteq S \cup \{z_1, z_2, z_3\} \cup A_{i_0}\} \cup \\
&\quad \{i_0, j_0\}, \text{ and} \\
\mathcal{Y} &= \{X_i \mid i \in V(T) \setminus \{i_0\}\} \cup \{Y_{i_0}, Y_{j_0}\},
\end{aligned}$$

where $Y_{i_0} = X_{i_0} \setminus S$ and $Y_{j_0} = \{z_1, z_2, z_3\} \cup S \cup A_{i_0}$. Notice that, as z_1, z_2, and z_3 induce a separating triangle in K' there is no $i \in V(T)$ such that $X_i \cap \{z_1, z_2, z_3\} \neq \emptyset$ and $X_i \cap [V(K') \setminus (S \cup \{z_1, z_2, z_3\})] \neq \emptyset$. This implies that $\mathcal{T}'$ is indeed a tree decomposition. It is also easy to verify that $\mathcal{T}'$ is small. Indeed, notice that neither $Y_{i_0} \subseteq Y_{j_0}$ nor $Y_{j_0} \subseteq Y_{i_0}$. Evenmore, for every $i \in V(T) \setminus \{i_0\}$, neither $X_i \subseteq Y_{j_0}$ nor $X_i \subseteq Y_{i_0}$, as this would imply $X_i \subseteq X_{i_0}$ which is a contradiction to the fact that $\mathcal{T}$ is small. Finally notice that if there exists an $i \in V(T) \setminus \{i_0\}$ such that $Y_{j_0} \subseteq X_i$, then $\{z, z_1, z_2, z_3\} \subseteq Y_{j_0} \subseteq X_i$, a contradiction to the hypothesis as then by the definition of a tree decomposition there exists a vertex $j \in I'$ such that X_j contains all the vertices of the clique.

It is also easy to see that for every $i \in V(T) \setminus \{i_0\}$, $Y_{i_0} \nsubseteq X_i$. Thus, $\mathcal{T}'$ contradicts to the choice of $\mathcal{T}$ (Condition (ii)). Therefore, there exists a vertex $j \in I'$, say j_0, such that $\{z, z_1, z_2, z_3\} \subseteq X_{j_0}$.

We will now prove that $\mathcal{T}$ does not satisfy Condition (iii). Similarly, as above, we modify $\mathcal{T}$ into a tree decomposition $\mathcal{T}''$ in the following way. We remove S from X_{i_0} and add it to X_{j_0}. Then, we remove all the edges between the neighbors $i \in I' \setminus j_0$ of i_0 in T, whose common vertices with X_{i_0} are either apices or vertices of S and the clique, and make them adjacent to j_0 instead.

Formally, let $\mathcal{T}'' = (\mathcal{Z}, T'')$ be the tree decomposition of G with

$$\begin{aligned}
V(T'') &= V(T),\\
E(T'') &= (E(T) \setminus \{\{i, i_0\} \mid i \in (N_T(i_0) \setminus \{j_0\})\\
&\qquad\qquad\qquad \wedge (X_i \cap X_{i_0}) \subseteq S \cup \{z_1, z_2, z_3\} \cup A_{i_0}\}) \cup\\
&\quad \{\{i, j_0\} \mid i \in N_T(i_0)\\
&\qquad\qquad \wedge (X_i \cap X_{i_0}) \subseteq S \cup \{z_1, z_2, z_3\} \cup A_{i_0}\}, \text{ and}\\
\mathcal{Z} &= \{X_i \mid i \in V(T) \setminus \{i_0, j_0\}\} \cup \{Z_{i_0}, Z_{j_0}\},
\end{aligned}$$

where $Z_{i_0} = X_{i_0} \setminus S$ and $Z_{j_0} = X_{j_0} \cup S$. It is again easy to see that $\mathcal{T}''$ is a small tree decomposition of G. Notice also that $\mathcal{T}$ and $\mathcal{T}''$ contain the same amount of bags. Furthermore, it is easy to see that

$$\sum_{\substack{D, D' \in \mathcal{Z}\\ D \neq D'}} |D \cap D'| < \sum_{\substack{L, L' \in \mathcal{X}\\ L \neq L'}} |L \cap L'|,$$

a contradiction to the choice of $\mathcal{T}$ (Condition (iii)). Therefore, K' does not contain any clique of size 4.

Recall now that for every $i \in \mathcal{I}'$, $F[V(K') \cap X_i] \subseteq K'$ is a clique. Thus, for every $i \in I'$, $|V(K') \cap X_i| \leq 3$. $\square$

Recall now that $\mathcal{T}_{i_0, I'}$ is the collection of connected components of $T \setminus i_0$ that contain vertices of I' and recall also that there exists at most one tree in $\mathcal{T}_{i_0, I'}$, say T', that contains a vertex i_1 with $w(i_1) > w(i_0)$. Let $\mathcal{W}' = \{W'_1, W'_2, W'_3, W'_4\}$ be the collection of vertex-disjoint subwalls of W' of height $f_4(h) \cdot k$ not meeting the vertices of $P^{(h)}_{k \cdot f_4(h)+2}$ and $P^{[v]}_{k \cdot f_4(h)+2}$ (see Figure 6.10).

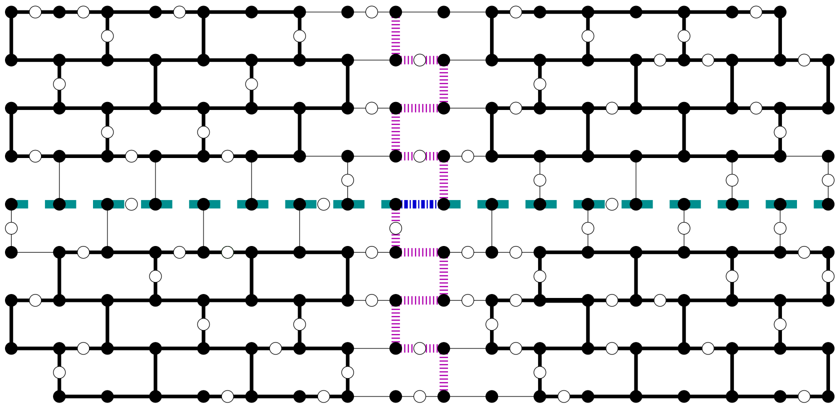


Figure 6.10: The paths $P^{(h)}_{k\cdot f_4(h)+2}$ (cyan - dashed) and $P^{[v]}_{k\cdot f_4(h)+2}$ (magenta - dotted) and the corresponding walls for $k = 1$ and $f_4(h) = 3$.

From Claim 9, X_{i_1} has at most 3 vertices in common with K', therefore there exists a subwall $\tilde{W} \in \mathcal{W}'$ of height $f_4(h) \cdot k$, with compass $\tilde{K}$ in F such that $V(\tilde{K}) \cap V(G_{T'}) = \emptyset$. (For the "big picture", see Figure 6.11.)

Consequently, if we set

$$\tilde{I} = \{i \in N_T(i_0) \mid X_i \cap V(\tilde{K}) \neq \emptyset\}$$

we have that $\tilde{I} \subseteq I' \setminus \{i_1\}$ and for every tree $\tilde{T} \in \mathcal{T}_{i_0,\tilde{I}} \subseteq \mathcal{T}_{i_0,I'} \setminus \{T'\}$ it holds that $\max\{w(i) \mid i \in V(\tilde{T})\} \leq f_3(h,k)$. Therefore, for every $\tilde{T} \in \mathcal{T}_{i_0,\tilde{I}}$, $\mathbf{tw}(\overline{G}_{\tilde{T}}) \leq f_3(h,k)$. As $G_{\tilde{T}}$ is a subgraph of $\overline{G}_{\tilde{T}}$, it follows that

$$\text{for every } \tilde{T} \in \mathcal{T}_{i_0,\tilde{I}}, \quad \mathbf{tw}(G_{\tilde{T}}) \leq f_3(h,k). \tag{6.2}$$

From Claim 9, it follows that for every $\tilde{T} \in \mathcal{T}_{i_0,\tilde{I}}$, the vertices in $V(G_{\tilde{T}}) \cap V(\tilde{K})$ induce a clique in $\tilde{K}$ with at most 3 vertices, where some of its edges may be virtual.

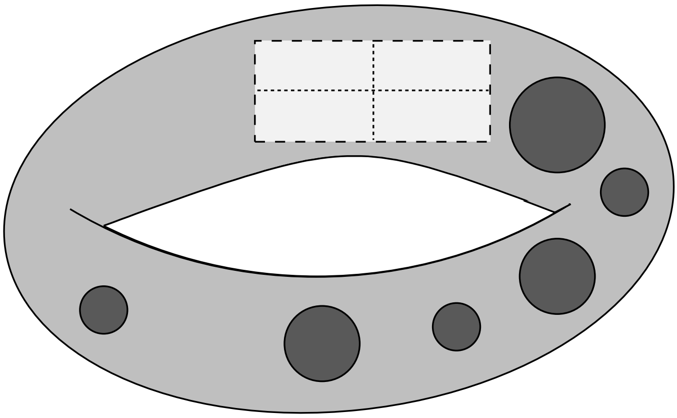


Figure 6.11: The compass $\tilde{K}$ of the wall $\tilde{W}$ is inside one of the four parallelograms belonging to the interior of the dashed parallelogram.

Let $\tilde{V} = V(F) \setminus V(\tilde{K})$ and $\mathcal{F}' = \{G_{\tilde{T}} \mid \tilde{T} \in \mathcal{T}_{i_0,\tilde{I}}\}$. Notice that $\tilde{K} = F \setminus \tilde{V}$. We denote by $\mathcal{F}$ the class $\mathcal{P}_{\mathcal{F}',\tilde{V},F}$. Our aim now is to use the graphs in $\mathcal{F}$ in order to define the compass of $\tilde{W}$ in the graph $\overline{G} \setminus A_{i_0}$ and construct the rural division of that compass.

Claim 10. *For every connected component Y of $T \setminus \{i_0\}$ that contains a vertex i_Y of $\tilde{I}$, there is a vertex in $G_Y \setminus X_{i_0}$ connected with $V(\tilde{K}) \cap V(G_Y)$ with $|V(\tilde{K}) \cap V(G_Y)|$ vertex-disjoint paths whose internal vertices belong to $G_Y \setminus X_{i_0}$.*

Proof of Claim 10. First, observe that, G_Y has at least one connected component G'_Y that contains the vertices in $V(\tilde{K}) \cap V(G_Y)$ and such that $V(G'_Y) \setminus X_{i_0} \neq \emptyset$. Otherwise, observe that we may safely remove the vertices in $V(\tilde{K}) \cap V(G_Y)$ from the bags X_i, $i \in V(Y)$ and end up to a contradiction in the choice of $\mathcal{T}$ (Condition (iii)). Evenmore, Condition (ii) implies that X_{i_0} is not a separator of G'_Y. Notice then that every vertex in $V(\tilde{K}) \cap V(G_Y)$ has a neighbor in $G'_Y \setminus X_{i_0}$, as if not, we would again end up to a contradiction in the choice of $\mathcal{T}$ (Condition (iii)). As G'_Y is connected there exists a vertex $v \in V(G'_Y) \setminus X_{i_0}$ and vertex-disjoint paths from v to the vertices of $V(\tilde{K}) \cap V(G_Y)$. $\square$

We call the edges in $\tilde{E} = E(\tilde{K}) \setminus E(G)$ *useless*. We also call all vertices in $V(\bigcup \mathcal{F}) \setminus V(\tilde{K})$ *flying* vertices. The non-flying vertices of a graph R in $\mathcal{F}$ are the *base* of R. Notice that, from the definition of $\mathcal{F}$, each graph R in $\mathcal{F}$ is a subgraph of the union of some graphs of $\mathcal{F}'$. From Observation 6.3 and (6.2), It follows that

(a) all graphs in $\mathcal{F}$ have treewidth at most $f_3(h, k)$

Observation 6.3 and Claim 9 yields that

(b) the base vertices of each R induce a clique of size 1,2, or 3 in $\tilde{K}$.

Also, from Claim 10 and the fact that $\tilde{V} \cup A_{i_0} \subseteq X_{i_0}$, we have that

(c) each pair of vertices of some graph in $\mathcal{F}$ are connected in G by a path whose internal vertices are flying.

Note that each clique mentioned in (b) may contain useless edges. Moreover, from (c), all virtual edges of $\tilde{K}$ are edges of such a clique. Let $\tilde{G} = (V(G), \tilde{E} \cup E(G))$, that is, we add in G all useless edges.

It now follows that $\tilde{G} \setminus A_{i_0}$ contains the wall $\tilde{W}$ as a subgraph and the compass of $\tilde{W}$ in $\tilde{G} \setminus A_{i_0}$ is

$$\tilde{K}^+ = \tilde{K} \cup \bigcup_{R \in \mathcal{F}} R$$

Notice that the wall $\tilde{W}$ remains flat in $\tilde{G}$. Indeed, suppose that Q_1 and Q_2 are two vertex-disjoint paths between the two anti-diametrical corners of $\tilde{W}$ such that the sum of their lengths is minimal. As not both of Q_1 and Q_2 may exist in $\tilde{K}$, some of them, say Q_1 contains some flying vertex. Let R be the graph in $\mathcal{F}$ containing that vertex. Then there are two vertices x and y of the base of R met by Q_1. From (b), $\{x, y\}$ is an edge of $\tilde{K}^+$ and we can substitute the portion of Q_1 that contains flying vertices by $\{x, y\}$, a contradiction to the minimality of the choice of Q_1 and Q_2.

Let $\tilde{E}^+ = E(\tilde{K}^+) \setminus E(\bigcup \mathcal{F})$, that is, $\tilde{E}^+$ is the set of edges of $\tilde{K}$ not contained in any graph R of $\mathcal{F}$. It follows that all useless edges are contained in $\tilde{E}^+$, that is,

$$\tilde{E} \subseteq \tilde{E}^+ \tag{6.3}$$

For every $e \in \tilde{E}^+$, we denote by $\tilde{G}_e$ the graph formed by the edge e (that is, the graph $\tilde{G}[e]$) and let $\mathcal{E} = \{\tilde{G}_e \mid e \in \tilde{E}^+\}$. We set $\tilde{\mathcal{D}}^+ = \mathcal{F} \cup \mathcal{E}$. Notice that,

$$\text{For every graph } R \in \mathcal{F}, \qquad \partial_{\tilde{K}^+} R \text{ is the base of } R \tag{6.4}$$

$$\text{For every graph } \tilde{G}_e \in \mathcal{E}, \qquad \partial_{\tilde{K}^+} \tilde{G}_e = V(\tilde{G}_e) \tag{6.5}$$

Claim 11. *$\tilde{\mathcal{D}}^+ = \mathcal{F} \cup \mathcal{E}$ is a rural division of $\tilde{K}^+$.*

Proof of Claim 11. Properties 1 and 2, follow from the construction of the graphs in $\mathcal{F}$ and $\mathcal{E}$. Moreover, Properties 3 and 4 follow from (c) and (b) respectively. For Property 5, recall that $\tilde{W}$ is a subwall of W' whose compass K' in F can be embedded in a closed disk Δ such that the perimeter of W' is identical to its boundary. This implies that $\tilde{K}$ can be embedded in a closed disk $\tilde{\Delta} \subseteq \Delta$ such that the corners c_1, c_2, c_3, and c_4 of $\tilde{W}$ appear in this order on its boundary. We now consider the following hypergraph:

$$\tilde{H}^+ = (\bigcup\{\partial_{\tilde{K}^+} D \mid D \in \tilde{\mathcal{D}}^+\}, \{\partial_{\tilde{K}^+} D \mid D \in \tilde{\mathcal{D}}^+\}).$$

Notice that $V(\tilde{H}^+) = V(\tilde{K})$. We can now construct $I(\tilde{H}^+)$ by applying, for each $D \in \tilde{\mathcal{D}}^+$, the following transformations on the planar graph $\tilde{K}$.

- If $|\partial_{\tilde{K}^+} D| = 1$, we add a new vertex and an edge that connects it with the unique vertex of $\partial_{\tilde{K}^+} D$.
- If $|\partial_{\tilde{K}^+} D| = 2$, we subdivide the edge of $\tilde{K}[\partial_{\tilde{K}^+} D]$ (recall that $\tilde{K}[\partial_{\tilde{K}^+} D]$ is isomorphic to K_2).
- If $|\partial_{\tilde{K}^+} D| = 3$, we apply a ΔY-transformation in $\tilde{K}[\partial_{\tilde{K}^+} D]$ (recall that $\tilde{K}[\partial_{\tilde{K}^+} D]$ is isomorphic to K_3).

From Observation 6.2 and Lemma 6.5, it follows that the obtained graph remains embedded in $\tilde{\Delta}$ (thus, it is also planar). It now remains to show that for each $e \in E(\tilde{H}^+)$ there exist $|e|$ vertex-disjoint paths between e and C in $\tilde{K}^+$. Notice that for each $e \in E(H^+)$ the vertices of e belong to $\tilde{K}$. Finally, there are $|e|$ paths between e and C, otherwise we would have a contradiction to the choice of the tree-decomposition. For this, notice that if there do not exist $|e|$ vertex-disjoint paths between e and C then there exists a separator of e and C of size strictly smaller that $|e|$. Then using similar arguments as in the proofs of Claim 9 and Claim 10 we end up contradicting the choice of $\mathcal{T}$. Therefore all conditions required for Claim 11 hold. $\square$

Our aim now is to find in $G \setminus A_{i_0}$ a flat subdivided wall $\widehat{W}$ of height $f_4(h) \cdot k$. From (b),(c), and (6.4), all the useless edges of $\tilde{K}$ are induced by the sets $\partial_{\tilde{K}^+} R$, $R \in \mathcal{F}$, where $\tilde{K}[\partial_{\tilde{K}^+} R]$ is isomorphic to either K_2

or K_3. Our next step is to prove that, in both cases, we may find a flat subdivided wall in $G \setminus A_{i_0}$ of height $f_4(h) \cdot k$ that does not contain any useless edges.

Case 1. $\tilde{K}[\partial_{\tilde{K}^+} R]$ is isomorphic to K_2. Then, from (c), there exists a path in R whose endpoints are the vertices of $\partial_{\tilde{K}^+} R$ and such that its internal vertices are flying.

Case 2. $\tilde{K}[\partial_{\tilde{K}^+} R]$ is isomorphic to K_3. Claim 10, combined with the facts that $\tilde{V} \cup A_{i_0} \subseteq X_{i_0}$ and that $\forall_{R \in \mathcal{F}}\ \partial_{\tilde{K}^+} R \subseteq X_{i_0}$, imply that there exists a flying vertex v_R in R and vertex-disjoint paths between v_R and the vertices of $\partial_{\tilde{K}^+} R$ whose internal vertices are also flying.

The above case analysis implies that for each $R \in \mathcal{F}$ the edge $\{x, y\}$ or the triangle with vertices $\{x, y, z\}$, induced by $\partial_{\tilde{K}^+} R$ may be substituted, using subdivisions or ΔY-transformations by a flying path between x and y or by three flying paths from a flying vertex v_R to x, y, and z respectively. As all edges of these paths are flying, they cannot be useless and therefore they exist also in $G \setminus A_{i_0}$. We are now in position to apply Observation 6.2 and Lemma 6.6 and obtain that $\tilde{G} \setminus A_{i_0}$ contains a flat subdivided wall $\widehat{W}$ of height $f_4(h) \cdot k$ such that

(I) $E(\widehat{W}) \cap \tilde{E} = \emptyset$ (recall that $\tilde{E}$ is the set of the useless edges) and

(II) $\widehat{W}$ is isomorphic to a subdivision of $\tilde{W}$.

Therefore, from (I), $\widehat{W}$ is a flat subdivided wall of height $f_4(h) \cdot k$ in $G \setminus A_{i_0}$.

Let $\tilde{C}$ and $\widehat{C}$ be the corners of $\tilde{W}$ and $\widehat{W}$ respectively. We denote by σ be the bijection from $\tilde{C}$ to $\widehat{C}$ induced by the isomorphism in (II). We also enhance σ by defining $\phi = \sigma \cup \{(x, x) \mid x \in V(\tilde{W}) \setminus C(\tilde{W})\}$.

Let $\widehat{K}$ be the compass of $\widehat{W}$ in $G \setminus A_{i_0}$. We claim that

$$\widehat{\mathcal{D}} = \{D \cap \widehat{K} \mid D \in \tilde{\mathcal{D}}^+\}$$

is a rural division of $\widehat{K}$. This is easy to verify in what concerns Properties (1–4). Property (5) follows from the observation that the mapping ϕ, defined above, is an isomorphism between $H_{\tilde{K}^+}$ and $H_{\widehat{K}}$.

So far, we have found a flat subdivided wall $\widehat{W}$ in $G \setminus A_{i_0}$ and a rural division of its compass $\widehat{K}$. As each flap in $\widehat{\mathcal{D}}$ is a subgraph of a flap in $\tilde{\mathcal{D}}^+$ we obtain that all flaps in $\widehat{\mathcal{D}}$ have treewidth at most $f_3(h,k)$. By applying Lemma 6.10 $|A_{i_0}| - \mathbf{an}(H) + 1$ times, it follows that there exists a set $A \subseteq A_{i_0}$, such that $|A| \leq \mathbf{an}(H) - 1$ and $G \setminus A$ contains a flat subdivided wall W of height k such that $W \subseteq \widehat{W}$. Moreover, $V(K) \cap A_{i_0} = \emptyset$, where K is the compass of W in $G \setminus A$. As above,

$$\mathcal{D} = \{D \cap K \mid D \in \widehat{\mathcal{D}}\}$$

is a rural devision of K, where all of its flaps have treewidth at most $f_3(h,k)$. The theorem follows as f_3 is a linear function of k. □

The following corollary gives a more precise description of the structure of apex minor free graphs.

Corollary 6.1. *There exists a computable function f such that for every two graphs H and G, where H is an apex graph, and every $k \in \mathbb{N}$ one of the following holds:*

1. $\mathbf{tw}(G) \leq f(h) \cdot k$, *where* $h = |V(H)|$.

2. H *is a minor of* G.

3. G *contains a flat subdivided wall* W *where*

   - W *has height* k *and*

- *the compass of W has a rural division $\mathcal{D}$ such that each internal flap of $\mathcal{D}$ has treewidth at most $f(h) \cdot k$.*

6.5 Tilings of the Plane

In this section we state three theorems that can be obtained from Theorem 6.1. Two of them are already known and we are going to use them in the chapter after the next one while the third can be used to tie all of them together through tilings.

However, before we move on to these theorems, let us first define some notions from the tilings of the plane that are necessary.

6.5.1 Regular Tilings of the Plane

A *tiling* of the plane $\mathcal{T}$ is a countable family of closed sets $\mathcal{T} = \{T_1, T_2, \ldots\}$ that cover the plane without gaps or overlaps. Explicitly, the union of the sets $T_1, T_2, \ldots$ is to be the whole plane and the interiors of the sets T_i are pairwise disjoint.

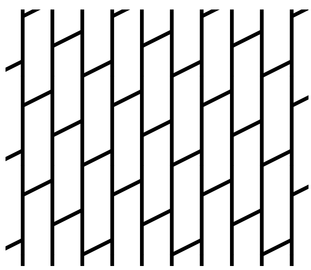


Figure 6.12: A monohedral tiling of the plane.

The tiles that we deal with in this section are topological disks and we only consider monohedral tilings, where a tiling is *monohedral* when all the tiles in it have the same size and shape. For an example of a monohedral tiling, see Figure 6.12. A monohedral tiling of the plane by a regular polygon is called *regular tiling* of the plane.

In the next subsection we are going to see that each one of the three theorems that we mentioned induce a different regular tiling of the plane, where one of them is self-dual and the other two are dual, in the graph-theoretic sense.

6.5.2 The Theorems and the Tilings

Let us now start stating the theorems.

Theorem 6.2 (Excluded Grid Theorem for Minors [53])**.** *There exists a computable function* $f : \mathbb{N} \to \mathbb{N}$ *such that for every two graphs* H *and* G*, and* $k \in \mathbb{N}$*, if* $\mathbf{tw}(G) \geq f(\mathbf{n}(H)) \cdot k$ *and* G *does not contain* H *as a minor then* G *contains the* $(k \times k)$*-grid as a minor.*

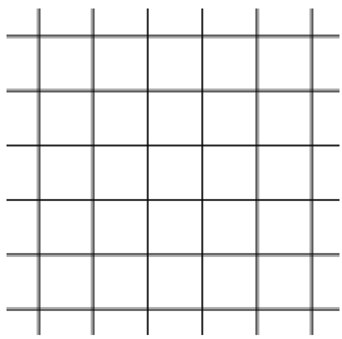


Figure 6.13: The square tiling of the plane.

It is very easy to see how this can be obtained from Theorem 6.1 as the $(k \times k)$-grid is a minor of a wall of height k. Notice now that the

$(k \times k)$-grid can be considered as the regular square tiling of the plane. (See Figure 6.13.)

Let us now go on to the next.

Theorem 6.3 (Excluded Grid Theorem for Contractions [83])**.** *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every two graphs H and G, where H is a connected apex graph, and $k \in \mathbb{N}$, if $\mathbf{tw}(G) \geq f(\mathbf{n}(H)) \cdot k$ and G does not contain H as a minor then G contains Γ_k as a contraction.*

As above it is fairly easy to see that Theorem 6.3 can be obtained as a corollary of Corollary 6.1. Let us again notice that the $(k \times k)$-triangulated grid can be consider as the regular triangular tiling of the plane. (See Figure 6.14.)

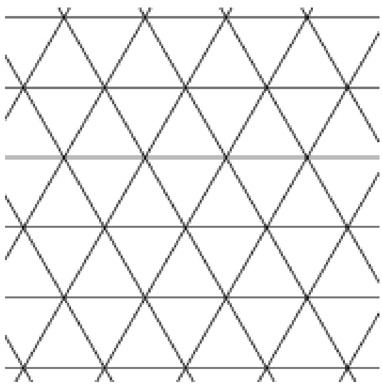


Figure 6.14: The regular triangular tiling of the plane.

Let us now move on to the last theorem of this section, which as above, asserts the existence of a large wall as a topological minor in a graph G of big enough tree-width that excludes a graph H as a minor.

Theorem 6.4 (Excluded Grid Theorem for Topological Minors)**.** *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every two graphs H and G, and $k \in \mathbb{N}$, if $\mathbf{tw}(G) \geq f(\mathbf{n}(H)) \cdot k$ and G does not contain H as a minor then G contains W_k as a topological minor.*

It is very easy to confirm that to aforementioned theorem can be obtained from Theorem 6.1. Notice now that, as above, the wall of height k can be considered as the regular hexagonal tiling of the plane, where the tiles are also named *honeycombs*. (See Figure 6.15.)

Let us notice here that while the honeycombs of the tiling and the bricks of the wall are equivalent in the topological sense, they are not equivalent to the geometric sense.

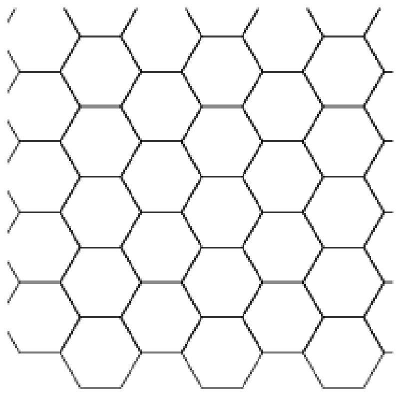


Figure 6.15: The regular hexagonal tiling of the plane.

Remark 3. Let us notice here that Theorem 6.3 differs from Theorems 6.2 and 6.4 in the following sense. While in Theorems 6.2 and 6.4 it is enough to exclude any graph H as a minor in order to ensure the existence of the grid and the wall as a minor and a topological minor, respectively, in order to assert the existence of the triangulated grid as a contraction, the restriction that H is an apex graph is necessary [83]. We would also like to comment here that in the chapter after the next we are going to use Theorem 6.4 and a variation of it in order to define the notion of bidimensionality on graph parameters that are closed under taking of (distance) topological minors.

It is known that, in the plane, the square tiling is self-dual in the topological sense and the hexagonal and the triangular tilings are dual. See, for example, Figure 6.16.

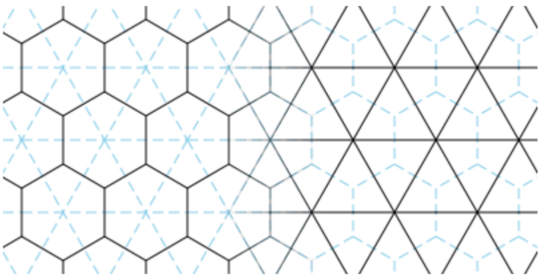


Figure 6.16: The duality of the triangular and the hexagonal tiling.

Let us now notice the following.

Observation 6.5. *Both the duality of the hexagonal and the triangular tiling and the self-duality of the square tiling can be "expanded" from the plane to the realm of the apex-minor-free graphs with large enough tree-width.*

CHAPTER 7

Excluding Immersions on Surface Embedded Graphs

In the previous chapter we saw how we can prove an optimized version of the Weak Structure Theorem, by building on the Graph Minors Strong Structure Theorem. As we have repeatedly seen the minor relation has been extensively studied (see, for example, [14, 25, 69, 105, 139, 193, 195, 197, 213, 219]).

However, the immersion ordering only recently attracted the attention of the research community [1, 17, 58, 101, 109, 119]. In [58], DeVos et al. proved that for every positive integer t, every simple graph of minimum degree at least $200t$ contains the complete graph on t vertices as a (strong) immersion and in [81] Ferrara et al., given a graph H, provide a lower bound on the minimum degree of any graph G in order to ensure that H is contained in G as an immersion. More recently, in [216], Seymour and Wollan proved a structure theorem for graphs excluding complete graphs as immersions.

In terms of graph colorings, Abu-Khzam and Langston in [1] provided evidence supporting the immersion ordering analog of Hadwiger's Conjecture, that is, the conjecture stating that if the chromatic number of a graph G is at least t, then G contains the complete graph on t vertices as an immersion, and proved it for $t \leq 4$. This conjecture is proven for $t = 5, 6$ and $t \leq 7$ by Lescure and Meyniel in [146] and by DeVos et al. in [59] independently. The most recent result on colorings is an approximation of the list coloring number on graphs excluding the complete graph as immersion [119].

Finally, in terms of algorithms, in [109], Grohe et al. gave a cubic time algorithm that decides whether a fixed graph H is contained in the input graph G as immersion.

In this chapter, inspired by the Graph Minors Weak Structure Theorem, we prove a structural characterization for the graphs that can be embedded in some surface of bounded genus. In particular, we show that if G is a graph that is embeddable in a surface of Eüler genus γ and H is a fixed graph then one of the following happens: Either G has bounded tree-width (by a function that depends only on γ and H), or G has "small" edge-connectivity (where the bound depends only on H), or G contains H as an immersion.

Let us now move to the following notions that are necessary for the proof of this result.

7.1 Necessary Notions

Walls continued. Let W be a wall of height k. We denote by L_i the i-th layer of W, $i \in \lceil \frac{k}{2} \rceil$. We also denote by A_i the annulus defined by the cycles L_i and L_{i+1}, that is, by the i-th and $(i+1)$-th layer, $i \in [\lceil \frac{k}{2} \rceil - 1]$. Given an annulus A defined by two cycles C_1 and C_2, we denote by $\overset{o}{A}$ the interior of A, that is, $A \setminus (C_1 \cup C_2)$.

A subdivided wall of height k is called *tight* if

1. the closed disk defined by the innermost ($\lceil \frac{k}{2} \rceil$-th) layer of W is edge-maximal (for reasons of uniformity we will denote this disk by $A_{\lceil \frac{k}{2} \rceil}$),
2. for every $i \in [\lceil \frac{k}{2} \rceil - 1]$ the annulus A_i is edge-maximal under the condition that A_{i+1} is edge-maximal.

Given a wall W and a layer L of W, different from the perimeter of W, let W' be the sub-wall of W with perimeter L. W' is also called the *sub-wall of W defined by L*. We call the following vertices, *important* vertices of L; The original vertices of W that belong to L and have degree 2 in the underlying non-subdivided wall of W' but are not the corners of W' (where we assume that W' shares the original vertices of W). See Figure 7.1.

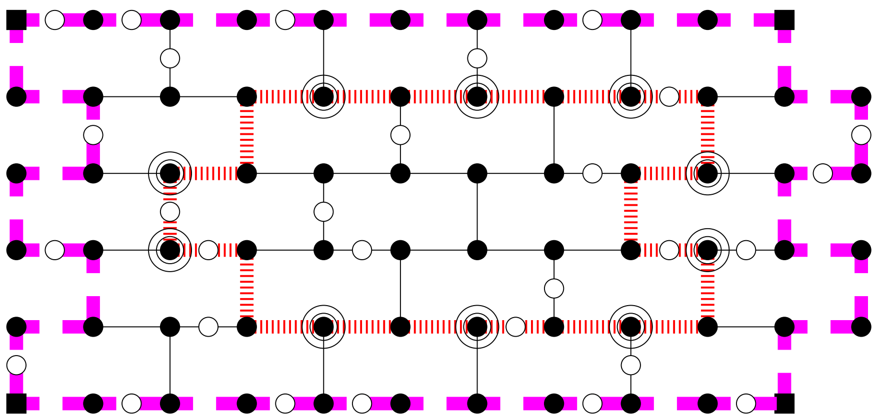


Figure 7.1: The important vertices the second layer of a wall of height 5.

Observation 7.1. *A layer L of a wall W that is different from its perimeter and defines a sub-wall W' of W of height k contains exactly $4k-2$ important vertices.*

From Lemma 6 in [83] and Lemma 6.3 we obtain the following.

Lemma 7.1. *Let G be a graph embedded in a surface of Eüler genus γ. If $\mathbf{tw}(G) \geq 48 \cdot (\gamma+1)^{\frac{3}{2}} \cdot (k+5)$ then G contains as a subgraph a subdivided wall of height k, whose compass in G is embedded in a closed disk Δ.*

Orthogonal drawings An *orthogonal drawing* of a graph G in a grid Γ is a mapping which maps

- vertices $v \in V(G)$ to sub-grids $\Gamma(v)$ (called *boxes*) such that for every $u_1, u_2 \in V(G)$ with $u_1 \neq u_2$, $\Gamma(u_1) \cap \Gamma(u_2) = \emptyset$, and
- edges $\{u_1, u_2\} \in E(G)$ to (u_1', u_2')-paths whose internal vertices belong to $\Gamma - \bigcup_{v \in V(G)} \Gamma(v)$, their endpoints u_i' (called *joining vertices of* $\Gamma(u_i)$) belong to the perimeter of $\Gamma(u_i)$, $i \in [2]$, and for every two disjoint edges $e_i \in E(G)$, $i \in [2]$, the corresponding paths are edge-disjoint.

It is known that.

Lemma 7.2 ([19])**.** *If G is a simple graph then it admits an orthogonal drawing in an $(\frac{m+n}{2} \times \frac{m+n}{2})$-grid. Furthermore, the box size of each vertex v is $\frac{\deg(v)+1}{2} \times \frac{\deg(v)+1}{2}$.*

7.2 Preliminary Combinatorial Lemmata

Detachment tree of $\mathcal{P}$ in u Let G be a graph embedded in a closed disk Δ, $v, v_1, v_2, \ldots, v_k$ distinct vertices of G and $\mathcal{P} = \{P_i \mid i \in [k]\}$ be a family of k confluent edge-disjoint paths such that P_i is a path from v to v_i, $i \in [k]$. Let $u \in V(G) \setminus \{v, v_i \mid i \in [k]\}$ belonging to more than one paths in $\mathcal{P}$. Assume that $\mathcal{P}_u = \{P_{i_1}, P_{i_2}, \ldots, P_{i_r}\}$ is the family of paths in $\mathcal{P}$ that contain u. Let Δ_u be a closed disk around u. We denote by $e^1_{i_r}$ and $e^2_{i_r}$ the edges of P_{i_j} incident to u, $j \in [r]$. Given any edge e with $u \in e$ we denote by u_e its common point with the boundary of Δ_u. We construct a tree T_u in the following way and call it, *the detachment tree of $\mathcal{P}$ in u*.

Consider the outerplanar[1] graph obtained from the boundary of Δ_u by adding the edges $\{u^1_{e_{i_j}}, u^2_{e_{i_j}}\}$, $j \in [r]$. We subdivide the edges $\{u_{e^1_{i_j}}, u_{e^2_{i_j}}\}$, $j \in [r]$, resulting to a planar graph. For every bounded face f of the graph, let $V(f)$ denote the set of vertices that belong to f. We add a new vertex v_f in its interior and we make it adjacent to the vertices of $(V(f) \cap \{u_e \mid e \in u\}) \setminus \{u^1_{e_{i_j}}, u^2_{e_{i_j}} \mid j \in [r]\}$. Finally we remove the edges whose both endpoints lie on the boundary of Δ_u. We denote this tree by T_u. Notice that for every e with $u \in e$, the vertex u_e is a leaf of T_u. (See Figure 7.2.)

We replace u by T_u in the following way. First we subdivide every edge $e \in G$ incident to u, and denote by u_e the vertex added after the subdivision of the edge e. We denote by G_s the resulting graph. Consider now the graph $G^r = (G^s \setminus u) \cup T_u$ (where, without loss of generality, we assume that $V(G \setminus u) \cap V(T_u) = \{u_e \mid u \in e\}$). The graph G^r is called *the graph obtained from G by replacing u with the detachment tree of $\mathcal{P}$ in u*.

Given a family of paths $\mathcal{P}$, we denote by $\mathcal{O}(\mathcal{P})$ the set of vertices in $\mathcal{P}$ that belong to more than one paths.

Observation 7.2. *Let k, h be positive integers and G be a multigraph*

[1] A graph is called *outerplanar* if it is a plane graph and all of its vertices lie in the outer face.

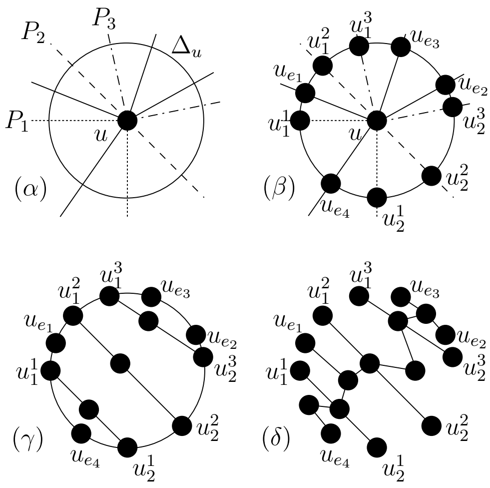


Figure 7.2: Example of the construction of a detachment tree.

containing as a subgraph a subdivided wall W of height h, whose compass C is embedded in a closed disk Δ. Furthermore, let v, v_i, $i \in [k]$, be vertices of W such that, there exists a confluent family $\mathcal{P}$ of k edge-disjoint paths from v to the vertices v_i, $i \in [k]$. Finally, let $u \in V(C) \setminus \{v, v_i \mid i \in [k]\}$ belonging to more than one paths of $\mathcal{P}$. The graph G^r obtained from G by replacing u with the detachment tree of $\mathcal{P}$ in u, T_u, contains as a subgraph a subdivided wall W' of height h, whose compass is embedded in Δ and there exists a family $\mathcal{P}'$ of k confluent edge-disjoint paths from v to v_i, $i \in [k]$, in W' such that $\mathcal{O}(\mathcal{P}') \subseteq \mathcal{O}(\mathcal{P}) \setminus \{u\}$.

Proof. Notice first that it is enough to prove the observation for the case where $u \in V(W)$. Let e_1, e_2 (and e_3) be the edges of W that are incident to u. Notice now that the vertices u_{e_1}, u_{e_2} (and u_{e_3}) are leaves of T_u. Thus, from a folklore result, there exists a vertex $u' \in V(T_u)$ such that there exist 2 (or 3) internally vertex-disjoint paths from u' to u_{e_1} and u_{e_2}

(and u_{e_3}). □

We now state the following auxiliary defitions. Let G be a multigraph that contains as a subgraph a wall of height k whose compass is embedded in a closed disk. Let $v \in A_{\lceil \frac{k}{2} \rceil}$, that is, let v be a vertex contained in the closed disk defined by the innermost layer of W. Let also P be a path from v to the perimeter of W. For each layer j of the wall, $2 \leq j \leq \lceil \frac{k}{2} \rceil$, we denote by x_P^j the first vertex of P (starting from v) that also belongs to L_j and we call it *incoming vertex of P in L_j*.

We denote by P^j the maximal sub-path of P that contains v and is entirely contained in the wall defined by L_j. We denote by y_P^j its endpoint in L_j and call it *outgoing vertex of P in L_j*. Notice that x_P^j and y_P^j are not necessarily distinct vertices.

Lemma 7.3. *Let l and k be positive integers, G be a graph, W be a tight subdivided wall of height k, that is a subgraph of G and whose compass is embedded in a closed disk Δ, and v be a vertex such that $v \in A_{\lfloor \frac{k}{2} \rfloor}$. If there exist l vertex-disjoint paths P_i, $i \in [l]$, from v to vertices of the perimeter then there is a brick B of W with $B \cap \overset{o}{A}_{j-1} \neq \emptyset$ that contains both $y_{P_i}^j$ and $x_{P_i}^{j-1}$.*

Proof. Assume the contrary. Then it is easy to see that we can construct an annulus A'_j such that $A_j \subsetneq A'_j$ and $|E(A_j)| < |E(A'_j)|$, a contradiction to the tightness of the wall. (See Figure 7.3.) □

Lemma 7.4. *Let k be a positive integer and G be a multigraph that contains as a subgraph a subdivided wall W of height at least $4 \cdot k^2 + 1$, whose compass K is embedded in a closed disk Δ. Let also V be a set of k vertices lying in the perimeter P of W, whose mutual distance in the underlying non-subdivided wall is at least 2. If there exist a vertex $v \in A_{2 \cdot k^2+1}$ and k*

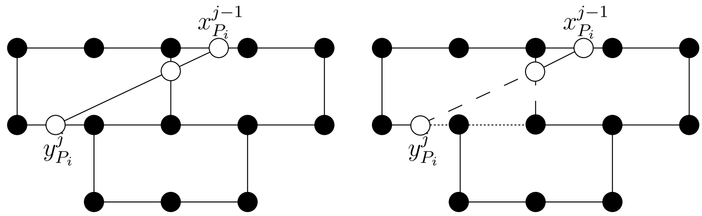


Figure 7.3: We replace the dotted line of the wall by the dashed line.

internally vertex-disjoint paths from v to vertices of P, then there exist k vertex-disjoint paths from v to the vertices of V in K.

Proof. Assume first, without loss of generality, that the wall W is tight. Let $P_1, P_2, \ldots, P_k$ be the paths from v to P and, without loss of generality, let $[P_1, P_2, \ldots, P_k, P_1]$ be the cyclic ordering according to which they appear in W clockwise. Our objective is to reroute the paths $P_i, i \in [k]$, so that they end up to the vertices of V. Let let P_i' be the sub-path of P_i that starts from v and stops the first time P_i meets the $(k+1)$-th layer of W, that is, $P_i' = P_i[v, x^{k+1}_{P_i}]$, $i \in [k]$.

We set $j_0 = k^2 + 1$. Consider the layer L_{j_0} and for every $i \in [k]$, let T_i denote the path of L_{j_0} starting from $x^{j_0}_i$ and ending in $x^{j_0}_{i+1}$ (considered clockwise), where in the case $i = k$ we abuse notation and assume that $x^{j_0}_{k+1} = x^{j_0}_1$ (see Figure 7.4). Let also $i_0 \in [k]$ be the index such that the path T_{i_0} contains the maximum number of important vertices amongst the T_i's. Without loss of generality, we may assume that $i_0 = \lceil \frac{k}{2} \rceil$. From Observation 7.1, as L_{j_0} defines a sub-wall W' of W of height $2 \cdot k^2 + 1$, L_{j_0} contains exactly $8 \cdot k^2 + 2$ important vertices. Thus, at least $7k$ important vertices are contained in the interior of T_{i_0}. Let $\{u_1, u_2, \ldots, u_k\}$ be a set of successive important vertices in T_{i_0} appearing in this order from $x^{j_0}_{i_0}$ to $x^{j_0}_{i_0+1}$, such that the paths $T_{i_0}[x^{j_0}_{i_0}, u_1]$ and $T_{i_0}[u_k, x^{j_0}_{i_0+1}]$ internally contain at least $3k$ important vertices.

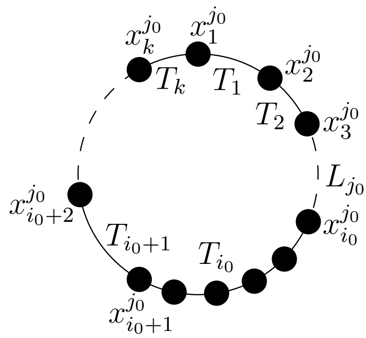


Figure 7.4: The paths T_i, $i \in [k]$.

Notice that, without loss of generality, we may assume that the vertices u_i, $i \in [k]$, belong to the northern path of W'. Recall here that each original vertex w of $W' \subseteq W \setminus P$ is contained in exactly one vertical path $P_w^{[v]}$ of W. For every $i \in [k]$ we assign a path R_i to the vertex u_i in the following way. Let R_i be the maximal subpath of $P_{u_i}^{[v]}$ that has u_i as an endpoint and does not contain any of the vertices belonging to the interior of the disk defined by L_{j_0} in the compass of W. Note here that, by the way they were defined, the paths R_i, $i \in [k]$, are vertex-disjoint (See Figure 7.5).

For every $i \in [k]$, we denote by u_i^f the important vertex of L_{k+1} that also belongs to R_i. It is easy to notice that in the annulus defined by L_{k+1} and L_1 there exist k vertex-disjoint paths from the vertices u_i^f, $i \in [k]$, to the vertices of V. In the remainder of the proof, using subpaths of the paths R_i, $i \in [k]$, and of the $\lceil \frac{k-2}{2} \rceil$ successive layers of W that precede L_{j_0} we reroute the paths P_i', $i \in [k]$, so that they end up to the vertices u_i^f, $i \in [k]$. Note that such a construction will complete the proof of the lemma.

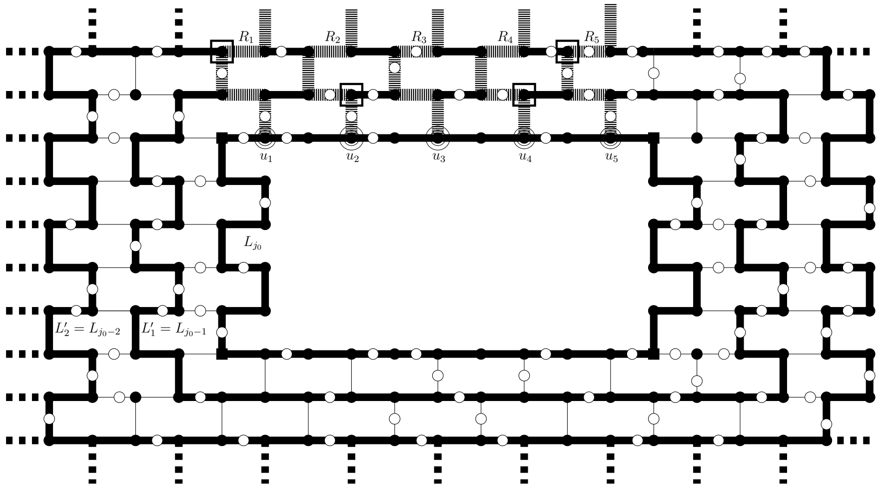


Figure 7.5: The important vertices of L_{j_0}, the layers L'_1 and L'_2, and the paths R_i.

First, notice that for k odd (respectively, even), L_{j_0} contains one (respectively, two vertex-disjoint) path F_1 (respectively, paths F_1 and F_2) from $x_{i_0}^{j_0}$ to u_{i_0} (respectively, from $x_{i_0}^{j_0}$ to u_{i_0} and from $x_{i_0+1}^{j_0}$ to u_{i_0+1}).

Consider now the $\lceil \frac{k-2}{2} \rceil$ successive layers of W preceding L_{j_0}, that is, the layers $L'_j = L_{j_0-j}$, $j \in [\lceil \frac{k-2}{2} \rceil]$. For every $j \in [\lceil \frac{k-2}{2} \rceil]$, let $u_{i_0-j}^{j}$ be the first time the path R_{i_0-j} meets L'_j when starting from u_{i_0-j}. Evenmore, in the case where k is odd (respectively, even), for every $j \in [\lceil \frac{k-2}{2} \rceil]$, let $u_{i_0+j}^{j}$ (respectively, $u_{i_0+1+j}^{j}$) be the first time the path R_{i_0+j} (respectively, R_{i_0+1+j}) meets L'_j starting from u_{i_0+j} (respectively, u_{i_0+1+j}). (See, for example, the vertices inside the squares in Figure 7.5.)

Towards our ultimate goal, we now need to prove the following.

Claim 12. *Let k be an odd (respectively, even) integer. For every $j \in [\lceil \frac{k-2}{2} \rceil]$, there exist two vertex-disjoint paths F_j^1 and F_j^2 be-*

tween the pairs of vertices $(x_{i_0-j}^{j_0-j}, u_{i_0-j}^{j})$ *and* $(x_{i_0+j}^{j_0-j}, u_{i_0+j}^{j})$ *(respectively,* $(x_{i_0+1+j}^{j_0-j}, u_{i_0+1+j}^{j})$*) that do not intersect the paths* $\{R_l \mid i_0 - j < l < i_0 + j\}$ *(respectively,* $\{R_l \mid i_0 - j < l < i_0 + 1 + j\}$*).*

Proof of Claim 12. Indeed, this holds by inductively applying the combination of Lemma 7.3 with the assertion that for every $j \leq 2 \cdot k^2 + 1$ and every p, q with $1 < p < q < k$, the outgoing vertices of P_{p-1} and P_{q+1} and the incoming vertices of P_p and P_q in the layer L_j, y_{p-1}^j, y_{q+1}^j, x_p^j, and x_q^j respectively appear in L_j respecting the clockwise order

$$[y_{p-1}^j, x_p^j, x_q^j, y_{q+1}^j]$$

in the tight wall W. $\square$

We recursively construct a family $\tilde{\mathcal{P}} = \{\tilde{P}_i \mid i \in [k]\}$ of paths from v to $\{u_i^f \mid i \in [k]\}$ in the following way.

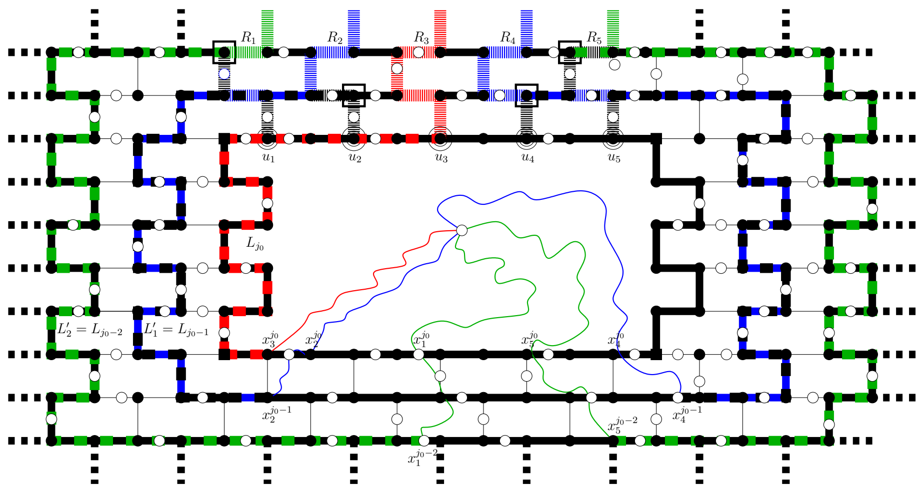


Figure 7.6: Part of the rerouted paths.

First we set,

$$\begin{aligned}\tilde{P}_{i_0} &= P_{i_0}\left[v, x_{i_0}^{j_0}\right] \cup F_1 \cup R_{i_0}\left[u_{i_0}, u_{i_0}^{f}\right]\\ \tilde{P}_{i_0+1} &= P_{i_0+1}\left[v, x_{i_0+1}^{j_0}\right] \cup F_2 \cup R_{i_0+1}\left[u_{i_0+1}, u_{i_0+1}^{f}\right]\end{aligned}$$

where $\tilde{P}_{i_0+1}$ is only considered when k is even. Then, for every $j \leq \lceil \frac{k-2}{2} \rceil$, we set

$$\tilde{P}_{i_0-j} = P_{i_0-j}\left[v, x_{i_0-j}^{j_0-j}\right] \cup F_j^1 \cup R_{i_0-j}\left[u_{i_0-j}, u_{i_0-j}^{f}\right]$$

Evenmore, for every $j \leq \lceil \frac{k-2}{2} \rceil$, we set

$$\tilde{P}_{i_0+j} = P_{i_0+j}\left[v, x_{i_0+j}^{j_0-j}\right] \cup F_j^2 \cup R_{i_0+j}\left[u_{i_0+j}^{j}, u_{i_0+j}^{f}\right]$$

in the case where k is odd, and

$$\tilde{P}_{i_0+1+j} = P_{i_0+1+j}\left[v, x_{i_0+1+j}^{j_0-j}\right] \cup F_j^2 \cup R_{i_0+1+j}\left[u_{i_0+1+j}^{j}, u_{i_0+1+j}^{f}\right]$$

in the case where k is even.

Claim 12 implies that the paths in $\tilde{\mathcal{P}} = \{\tilde{P}_i \mid i \in [k]\}$ are vertex-disjoint. This completes the proof of the lemma. For a rough estimation of the position of the paths in the wall see Figure 7.6. □

We now prove the main result of this section.

Lemma 7.5. *Let k be a positive integer and G be a k-edge-connected multigraph embedded in a surface of Eüler genus γ that contains a subdivided wall W of height at least $4 \cdot k^2 + 1$ as a subgraph, whose compass C is embedded in a closed disk Δ. Let also S be a set of vertices in the perimeter of W whose mutual distance in the underlying non-subdivided wall is at least 2. If $|S| \leq k$ then there exist a vertex v in W and $|S|$ edge-disjoint paths from v to the vertices of S.*

Proof. Let $v \in A_{2k^2+1}$ and $u \in L_1$ be vertices belonging to the closed disk defined by the layer $L_{2 \cdot k^2+1}$ and to the perimeter of the wall respectively. As G is k-edge-connected there exist k edge-disjoint paths $P_1, P_2, \ldots, P_k$ connecting v and u. By Proposition 3.1, we may assume that the paths are confluent. Let $\mathcal{P}' = \{P'_i \mid i \in [k]\}$ be the family of paths $P'_i = P_i[v, x_i^1]$, $i \in [k]$.

Let also V be the set of vertices in $V(C) \setminus (V(L_1) \cup \{v\})$ that are contained in more than one path in $\mathcal{P}'$. We obtain the graph $\hat{G}$ by inductively replacing every vertex $z \in V$ with the detachment tree of $\mathcal{P}'$ in z. From Observation 7.2, $\hat{G}$ contains a wall $\hat{W}$ of height $4 \cdot k^2 + 1$ whose compass is embedded in Δ. Notice also that, as no changes have occurred in the perimeter of W, W and $\hat{W}$ share the same perimeter. Furthermore, $\hat{W}$ contains k internally vertex-disjoint paths from v to the perimeter of $\hat{W}$. Thus, from Lemma 7.4, $\hat{W}$ contains k vertex-disjoint paths from v to S. It is now easy to see, by contracting the trees T_z, $z \in V(C) \setminus (V(L_1) \cup \{v\})$, that W contains k edge-disjoint paths from v to S. □

7.3 Main Theorem

We now state and prove the main theorem of this chapter.

Theorem 7.1. *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph G embedded in a surface of Eüler genus γ and every graph H one of the following holds:*

1. $\mathbf{tw}(G) \leq f(\gamma) \cdot \lambda \cdot k$, *where $\lambda = \Delta(H)$ and $k = \mathbf{m}(H)$.*

2. *G is not λ-edge-connected.*

3. $H \leq_{im} G$.

Proof. Let

$$f(\gamma,\lambda,k) = 48\cdot(\gamma+1)^{\frac{3}{2}}\cdot\left(\frac{4\,(4\lambda+1)\,k}{2}+5\right),$$

and assume that $\mathbf{tw}(G) \geq f(\gamma,\lambda,k)$ and that G is λ-edge-connected. From Lemma 7.1, we obtain that G contains as a subgraph a subdivided wall W of height $2\cdot(2\lambda+1)k$ whose compass is embedded in a closed disk.

In what follows, using this wall, we will construct a model of H in the compass of the wall. From Lemma 7.2, H admits an a orthogonal drawing ψ in an

$$\left(\frac{\mathbf{m}(H)+\mathbf{n}(H)}{2}\times\frac{\mathbf{m}(H)+\mathbf{n}(H)}{2}\right)\text{-grid},$$

where the box size of each vertex $v \in V(H)$ is

$$\frac{\deg(v)+1}{2}\times\frac{\deg(v)+1}{2}.$$

Notice now that ψ can be scaled to an orthogonal drawing ϕ to the grid Γ of size

$$\left(\frac{2\,(4\lambda+1)\,(\mathbf{m}(H)+\mathbf{n}(H))}{2}+1\right)\times 2\left(\frac{2\,(4\lambda+1)\,(\mathbf{m}(H)+\mathbf{n}(H))+2}{2}+1\right),$$

where the box size of each vertex is

$$(4(\deg(v))^2+2)\times 2(4(\deg(v))^2+2),$$

the joining vertices of each box have mutual distance at least 2 in the perimeter of the box and no joining vertex is a corner of the box.

Evenmore, for every vertex u, $u \in \mathbf{Im}(\phi)\setminus\cup_{v\in V(H)}\Gamma(v)$ of degree 4, that is, for every vertex in the image of ϕ that is contained in the intersection of two paths, there is a box in the grid of size $(4\deg(u)^2+2)\times 2(4\deg(u)^2+2)$, denoted by $Q(u)$, containing only this vertex and vertices of the paths it belongs to. We denote by u^i, $i\in[4]$, the vertices of $\mathbf{Im}(\phi)$ belonging to the boundary of $Q(u)$ and, for uniformity, also call them *joining vertices of* $Q(u)$.

Towards finding a model of H in the compass of the wall let us observe that the grid Γ contains as a subgraph a wall of height

$$(4\lambda + 1)\,(\mathbf{m}(H) + \mathbf{n}(H))$$

such that each one of the boxes, either $\Gamma(v)$, $v \in V(H)$, or $Q(v)$, where v is the intersection of two paths in the image of ϕ contains a wall $W(v)$ of height $4\deg(v)^2 + 1$ and the joining vertices of $\Gamma(v)$ (the vertices v^i, $i \in [4]$, respectively) belong to the perimeter of the wall and have distance at least 2 in it.

Consider now the mapping of H to W where the boxes $\Gamma(v)$ and $Q(v)$ are mapped into sub-walls $W(v)$ of W of height $4\deg(v)^2 + 1$ joined together by vertex-disjoint paths as induced by the orthogonal drawing ϕ. From Lemma 7.5, as every $W(v)$ has height $4\deg(v)^2 + 1$ and its compass is embedded in a closed disk, there exist a vertex $z_v \in V(W(v))$ and $\deg(v)$ edge-disjoint paths from z_v to the joining vertices of $W(v)$. It is now easy to see that the compass of W, and thus G, contains a model of H. $\square$

An immediate corollary of Theorem 7.1 is that.

Corollary 7.1. *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph G embedded in a surface of Eüler genus γ and every $k \in \mathbb{N}$ one of the following holds:*

1. $\mathbf{tw}(G) \leq f(\gamma) \cdot k^3$.

2. *G is not k-edge-connected.*

3. $K_{k+1} \leq_{im} G$.

Let us also notice that in the case where the graph H is the $(k \times k)$-grid then Theorem 7.1 impies that.

Corollary 7.2. *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph G embedded in a surface of Eüler genus γ and every $k \in \mathbb{N}$ one of the following holds:*

1. $\mathbf{tw}(G) \leq f(\gamma) \cdot k^2$.

2. *G is not 4-edge-connected.*

3. *$(k \times k)$-grid is an immersion of G.*

However, a straightforward argument shows that.

Theorem 7.2 (Excluded Grid Theorem for Immersions)**.** *There exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph G that is embedded in a surface of Eüler genus γ and every $k \in \mathbb{N}$ one of the following holds:*

1. $\mathbf{tw}(G) \leq f(\gamma) \cdot k$.

2. *G is not 4-edge-connected.*

3. *$(k \times k)$-grid is an immersion of G.*

Proof. Let

$$f(\gamma, k) = 48 \cdot (\gamma + 1)^{\frac{3}{2}} \cdot ((4^3 + 3) \cdot k + 5).$$

Assume that G is 4-edge-connected and that $\mathbf{tw}(G) \geq f(\gamma, k)$. From Lemma 7.1, since $\mathbf{tw}(G) \geq f(\gamma, k)$, it follows that G contains as a subgraph a subdivided wall W of height $(4^3 + 3)k$, whose compass in G is embedded in a closed disk Δ.

Consider the k^2 subwalls of W of height $(4^3 + 1)$ that occur after removing from it the paths $P^{[v]}_{(4^3+3)j}$, $P^{[h]}_{(4^3+3)j}$, $i, j \in [k]$. For every $i, j \in [k]$, we denote by $W_{(i,j)}$ the subwall that is contained inside the disk that is defined by the paths $P^{(h)}_{(4^3+3)(i-1)}$, $P^{(h)}_{(4^3+3)i}$, $P^{[v]}_{(4^3+3)(j-1)}$, and $P^{[v]}_{(4^3+3)j}$. In the case where $j = 1$ and $i = 1$, we abuse notation and consider as $P^{(h)}_{(4^3+3)(j-1)}$ and $P^{[v]}_{(4^3+3)(j-1)}$ the paths $P^{(h)}_1$ and $P^{[v]}_1$, respectively.

From Lemma 7.5 and the hypothesis that G is 4-edge-connected, for $k = 4$, it follows that in the compass of each one of the subwalls $\{W_{(i,j)} \mid i, j \in [k]\}$ we may find a vertex $v_{(i,j)}$ and four edge-disjoint paths from $v_{(i,j)}$ to the vertices $v^n_{(i,j)}$, $v^s_{(i,j)}$, $v^w_{(i,j)}$, and $v^e_{(i,j)}$, that lie in the northern, southern, western, and eastern path of the wall, respectively.

Finally, we consider the function $g((i, j)) = v_{(i,j)}$ that maps the vertex (i, j) of the $(k \times k)$-grid to the vertex $v_{(i,j)}$ of the wall $W_{(i,j)}$. Is now easy to see that g is an immersion model of the $(k \times k)$-grid in the compass of the wall W and the theorem follows as f is linear on k. $\square$

Theorem 7.2 asserts that every graph that is embedded in a surface of Eüler genus γ, has large enough tree-width, and large enough edge-connectivity contains a large enough grid as an immersion. The theorems that ensure the existence of large grids in graphs are called excluded grid theorems and, as we will see in the next chapter, play an important role in Bidimensionality Theory.

CHAPTER 8

Parameterized Complexity and Bidimensionality Theory

8.1 Introduction to Parameterized Complexity

As we already mentioned in Chapter 5, the meta-algorithm obtained from the results of P. Seymour and N. Robertson that recognizes, in cubic time, the classes of graphs that are closed under taking of minors had as a consequence the observation that the NP-complete problems have "different hardness levels".

Let us consider, for example, the problems k-Vertex Cover and k-Coloring.

k-VERTEX COVER	
Input:	A graph G and an integer k.
Parameter:	k.
Question:	Is there a set $S \subseteq V(G)$ with $\|S\| \leq k$ such that the graph $G \setminus S$ does not contain any edges?

k-COLORING	
Input:	A graph G and an integer k.
Parameter:	k.
Question:	Is there a proper coloring of the vertices of G using k colors?

As the class of graphs that have a vertex cover of size k is closed under taking of minors there exists a cubic time algorithm that recognizes it (and the hidden constants only depend on the class, that is, on the integer k).

In contrast to k-VERTEX COVER, a polynomial time algorithm for the problem of k-COLORING is not to be expected when k is greater or equal to 3 as it is known that 3-COLORING is an NP-complete problem and such a result would mean that $\mathsf{P} = \mathsf{NP}$.

Let us then observe that, the the dependence of the complexity of these problems on the the integer k is crucially different. The goal of parameterized complexity is to find ways of solving NP-hard problems more efficiently than brute force. In particular, the aim is to restrict the combinatorial explosion to a parameter that is hopefully much smaller than the input size.

Formally, given an alphabet Σ, a *parameterization* of Σ^* is any function $\kappa : \Sigma^* \to \mathbb{N}$. Then, for every problem $L \subseteq \Sigma^*$, the pair (L, κ), where κ is a parameterization of Σ^*, is called a *parameterized problem*. In order to distinguish between parameterized and classical problems we will use the prefix p-, so, for example, the problem VERTEX COVER in its parameterized form will be written as p-VERTEX COVER.

Let us notice here that the parameterization κ of a problem may be any function. The understanding of the dependence of the complexity of a parameterized problem on its parameterization constitutes one of the most important study subjects of Parameterized Complexity. (For more details regarding the parameterization of a problem see [77].)

The *natural parameterization* is the one in which the parameterization κ maps Σ to the desired solution size. For example then natural parameterization for p-VERTEX COVER is its size.

As it is known, in Classical Complexity there is a series of successive inclusions of complexity classes. The most important ones of them are:

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXPTIME}.$$

Many complexity classes, from the scope of Parameterized Complexity, and their inclusions are depicted below:

$$\mathsf{FPT} \subseteq \mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \cdots \subseteq \mathsf{W[P]} \subseteq \mathsf{\mathit{X}P}.$$

It is also conjectured that these inclusions are strict. Let (I, k) be an instance of a parameterized problem. We say that this problem belongs to the complexity class FPT, or is *fixed-parameter tractable*, if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is an arbitrary computable function depending on the parameter k only.

For example, from the meta-algorithm of N. Robertson and P. Seymour, we may obtain that p-VERTEX COVER belongs to FPT. The following simple recursive algorithm solves p-VERTEX COVER (where the parameter is the solution size) in $2^k \cdot n$ time.

1. If $E(G) = 0$, then return YES.

2. If $k = 0$, then return NO.

3. Choose and edge $e = \{u, v\}$ of the graph.

4. Recursively call the algorithm for $(G - u, k - 1)$ and $(G - v, k - 1)$ and return YES if at least one of these inputs returns YES.

The technique used in this algorithm is called *bounded-depth search tree* technique. For more on this technique see, for example [15, 30, 32, 80, 84, 107, 173–175]. For more on algorithmic techniques, see [11, 34, 47, 89, 108, 128, 148, 184, 215].

In the context of Parameterized Complexity Theory, the complexity class of fixed parameter tractable problems is equivalent to the class of the polynomial time solvable problems in Classical Complexity.

Similarly to the polynomial time and logarithmic space reductions of Classical Complexity, we may define FPT reductions in Parameterized Complexity. In particular, let us consider two parameterized problems (L, κ) and (L', κ') on the alphabets Σ and Σ' respectively. A function $R : \Sigma^* \to \Sigma'^*$ is called an FPT *reduction* of the problem (L, κ) to the problem (L', κ') if the following hold:

1. For every $x \in \Sigma^*$, x belongs to L if and only if $R(x)$ belongs to L',

2. The problem of computing R belongs to FPT with parameter $\kappa(x)$, that is, it can be computed in time $f(\kappa(x)) \cdot |x|^{O(1)}$, where f is a computable function, and

3. There exists a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for every $x \in \Sigma^*$, $\kappa'(R(x)) \leq g(\kappa(x))$.

It is easy to see that the complexity class FPT is closed under FPT reductions. The notions of hard and complete problems are defined in a similar way as in Classical Complexity if, instead of polynomial time reductions, we consider FPT reductions.

Just as NP-hardness is used as evidence that a problem probably is not polynomial time solvable in terms of Classical Complexity, the proof that a problem is hard for some of the complexity classes $\mathsf{W[1]}$, $\mathsf{W[2]}$, $\ldots$, $\mathsf{W[P]}$, and XP gives evidence that the problem is unlikely to be fixed-parameter tractable.

The main classes in this hierarchy are: The principal analogue of the classical intractability class NP is $\mathsf{W[1]}$, which is a strong analogue, because a fundamental problem complete for $\mathsf{W[1]}$ is the k-Step Halting Problem for Nondeterministic Turing Machines (with unlimited nondeterminism and alphabet size) — this completeness result provides an analogue of Cook's Theorem in Classical Complexity. In particular this means that an FPT algorithm for any $\mathsf{W[1]}$-hard problem would yield a $O(f(k)n^c)$ time algorithm for k-Step Halting Problem for Nondeterministic Turing Machines. A convenient source of $\mathsf{W[1]}$-hardness reductions is provided by the result that k-Clique is complete for $\mathsf{W[1]}$. The problem k-Dominating Set is complete for $\mathsf{W[2]}$ and XP is the class of all problems that are solvable in time $O(n^{g(k)})$. Finally, we should not neglect to mention the class $\mathsf{para\text{-}NP}$ which consists of all the problems that can be solved in FPT time by a non-deterministic Turing Machine. For an extensive introduction to Parameterized Complexity, see [66, 82, 172].

8.2 Distance Topological Minors

8.2.1 Spanners on Graphs

In [67], F. Dragan, F. Fomin, and P. Golovach proved that the problem k-Tree-width t-Spanner is fixed parameter tractable when we are restricted to the class of graphs that exclude some apex graph H as a minor.

For this proof they first showed that.

Theorem 8.1 ([67])**.** *Let G be a planar graph and S be a t-spanner of G. If G has tree-width k then S has tree-width $\Omega(\frac{k}{t})$.*

Then, they used the idea of its proof in order to show the more general combinatorial result.

Theorem 8.2. *Let H be an apex graph. If S is a t-spanner of a graph G that does not contain H as a minor, then S has tree-width $\Omega(\mathbf{tw}(G))$ (where the hidden constants in Ω depend only on the size of H and on t).*

The proof of the theorem above is based on the Theorem of E. Demaine and M. Hajiaghayi which ensures the existence of a large enough grid in a graph G that excludes some graph H as a minor and has large enough tree-width (see Theorem 6.2).

The proof of Theorem 8.2, however, occupies several pages and needs to deepen in the Graph Minors Theory. In this section we show how we may extend Theorem 6.1 in order to prove Theorem 8.2 in a simpler way. Let us start we the notion of the t-spanner.

Let t be a positive integer, G be a graph, and S be a spanning subgraph of G. We say that S is a (*multiplicative*) *t-spanner of G* if for every $x, y \in V(G)$, $\mathbf{dist}_S(x, y) \leq t \cdot \mathbf{dist}_G(x, y)$. We call t the *stretch factor of S*.

We also say that a graph H is a *distance topological minor* of a graph G if $H \leq_{tm} G$ and for every $u, v \in V(H)$, it holds that $\mathbf{dist}_H(u, v) \leq \mathbf{dist}_G(u, v)$.

We will now prove two lemmata which, when combined with Corollary 6.1, permit us to find the wall of height k as a distance topological minor in a graph G that does not contain a fixed apex graph H as a minor and has large enough tree-width.

Lemma 8.1. *Let k and d be positive integers, G be a graph that contains a flat subdivided wall W of height k as a subgraph, and $x, y \in V(W)$. If there exist d vertex-disjoint cycles $C_i, i \in [d]$, in the compass of W separating x and y such that $C_i \neq P$, $i \in [d]$, where P is the perimeter of W, then $\mathbf{dist}_G(x, y) > d$.*

Proof. First, recall that, we denote by c_i, $i \in [4]$, the corners of the wall according to the order they occur in P, that is, the pairs $\{c_1, c_3\}$ and $\{c_2, c_4\}$ are the pairs of the anti-diametrical corners.

Assume now, in contrary, that there exist d distinct cycles C_i, $i \in [d]$, in the compass of W separating x, y but $\mathbf{dist}_W(x, y) \leq d$. Let Q be a shortest path with endpoints x and y. As the distance between x and y is at most d and Q is a shortest path joining them, then Q contains at most $d - 2$ internal vertices. Thus, there exists an $i \in [d]$ such that the cycle C_i and the path Q are vertex-disjoint, that is, $V(C_i) \cap V(Q) = \emptyset$. This implies that we may find two vertex-disjoint paths with endpoints c_1, c_3 and c_2, c_4. This contradicts to the hypothesis that W is flat. Therefore, the lemma holds. □

Lemma 8.2. *Let k be a positive integer. If G is a graph that contains a flat subdivided wall W of height $6 \cdot k + 1$ as a subgraph, then G contains W_k as a distance topological minor.*

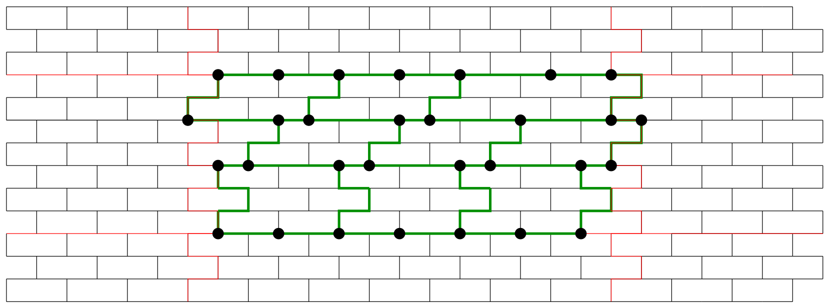


Figure 8.1: The wall W_3 as a distance topological minor in a flat wall of height 13.

Proof. Recall that W contains $6k+1$ vertical and $6k+1$ horizontal paths, $P_1^{[v]}, P_2^{[v]}, \ldots, P_{6k+2}^{[v]}$ and $P_1^{(h)}, P_2^{(h)}, \ldots, P_{6k+2}^{(h)}$. Let us consider the subwall of W whose perimeter is the cycle defined by the paths $P_{k+1}^{[v]}$, $P_{5k+1}^{[v]}$, $P_{k+1}^{(h)}$, $P_{5k}^{(h)}$. This is a wall of height $2k+1$. It is now easy to see, from Lemma 8.1, that inside this wall we may find the wall of height k as a distance topological minor. (See, for example, Figure 8.1.) $\square$

We are now able to prove the following.

Theorem 8.3 (Excluded Grid Theorem for Distance Topological Minors)**.** *There exists a computable function f such that for every two graphs H and G, where H is an apex graph, and $k \in \mathbb{N}$, if $\mathbf{tw}(G) \geq f(\mathbf{n}(H)) \cdot k$ and G does not contain H as a minor, then G contains W_k as a distance topological minor.*

Proof. Assume that G does not contain H as a minor and that $\mathbf{tw}(G) \geq f(\mathbf{n}(H)) \cdot (6k+1)$, where f is the function from Corollary 6.1. Then, from Corollary 6.1, G contains as a subgraph a flat subdivided wall of height $6 \cdot k + 1$. Thus, from Lemma 8.2, G contains the wall of height k as a distance topological minor. □

By following the proof of Theorem 8.1 in [67] and applying this more general theorem, it is easy to see that, we may obtain Theorem 8.2.

8.2.2 Duality completed ...

As observed in Remark 3, the Theorems 6.3 and 6.4 extend the duality of the regular triangular and hexagonal tilings in the realm of the graphs that exclude a fixed apex graph H as a minor and have large enough tree-width. We would like to also observe here the following.

Remark 4. The relations of contractions and distance topological minors are dual.

8.3 Bidimensionality Theory and Subexponential Algorithms

8.3.1 Bidimensionality Theory for Contractions and Minors

As we discussed in the previous section, the aim of Parameterized Complexity is the better understanding of the dependence of the complexity of the "hard problems" on the parameter k. In this section we exclusively deal with problems that are fixed parameter tractable, that is, they are solvable in time $f(k) \cdot n^{O(1)}$, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function, and more specifically we deal with f.

From the results in [28, 82, 126] there is evidence according to which when the input of a parameterized problem, where k is the parameter, is any general graph then the construction of an FPT algorithm for the problem with $f(k) = 2^{o(k)}$ is not expected. Nevertheless, after a series of results, is started to become apparent that for many problems, if the input of the algorithm is restricted on special graph classes, such as, for example, the class of planar graphs, then these are solved by an FPT algorithm with $f(k) = 2^{o(k)}$ (see, for example, [6, 7, 48–50, 52–55, 64, 87, 92]). Such an FPT algorithm is called *subexponential*.

Bidimensionality Theory provides a general technique according to which we may ensure the existence of subexponential algorithms when two specific conditions, a mathematical one and an algorithmic one, are met. The special interest in Bidimensionality Theory lies in the fact that, even though its results are mainly algorithmic, it is based on fundamental mathematical theorems of Graph Minors.

A special case of the theorem that consists the fundamental mathematical cornerstone in Bidimensionality Theory is the following:

Theorem 8.4 ([197]). *Let $k \in \mathbb{N}$. If G is a planar graph that does not contain the $(k \times k)$-grid as a minor then $\mathbf{tw}(G) \leq 6k - 5$.*

We call a parameter P *bidimensional under taking of minors with density δ* if

1. P is closed under taking of minors and
2. If R is the $(r \times r)$-grid then $P(R) = (\delta r)^2 + o((\delta r)^2)$.

For example, in Figure 8.2 one may see that vertex cover is bidimensional with density $\frac{1}{\sqrt{2}}$.

Similarly, we call a parameter P *bidimensional under taking of contractions with density δ* if

1. P is closed under taking of contractions and
2. If Γ_k is the $(r \times r)$-triangulated grid then $P(\Gamma_k) = (\delta r)^2 + o((\delta r)^2)$.

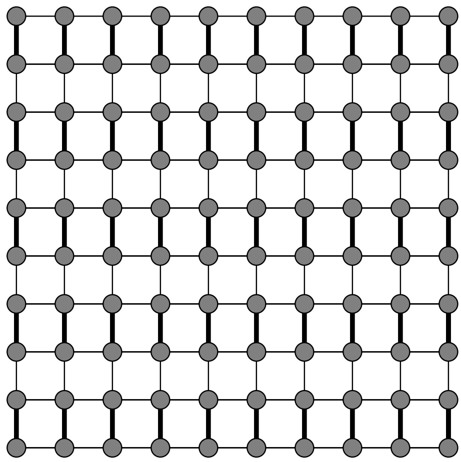


Figure 8.2: The bidimensionality of vertex cover.

Bidimensionality Theory is applied on graph parameters that are closed under taking of minors or contractions and such that, for the parameterized problem of computing the parameter, parameterized by the tree-width of the input graph, there exists an FPT algorithm with running time $2^{O(\mathbf{tw}(G))}n^{O(1)}$. (See Figure 8.3.)

Consider, for example, the following parameterization of p-VERTEX COVER.

p-VERTEX COVER	
Input:	A graph G and an integer k.
Parameter:	$\mathbf{tw}(G)$.
Question:	Is there a subset $S \subseteq V(G)$ with $\lvert S\rvert \leq k$ such that the graph $G \setminus S$ does not contain any edges?

According to results of F. Dorn in [65] there exists an algorithm that solves the problem in $2^{O(\mathbf{tw}(G))}n^{O(1)}$ time. Notice however that the fact

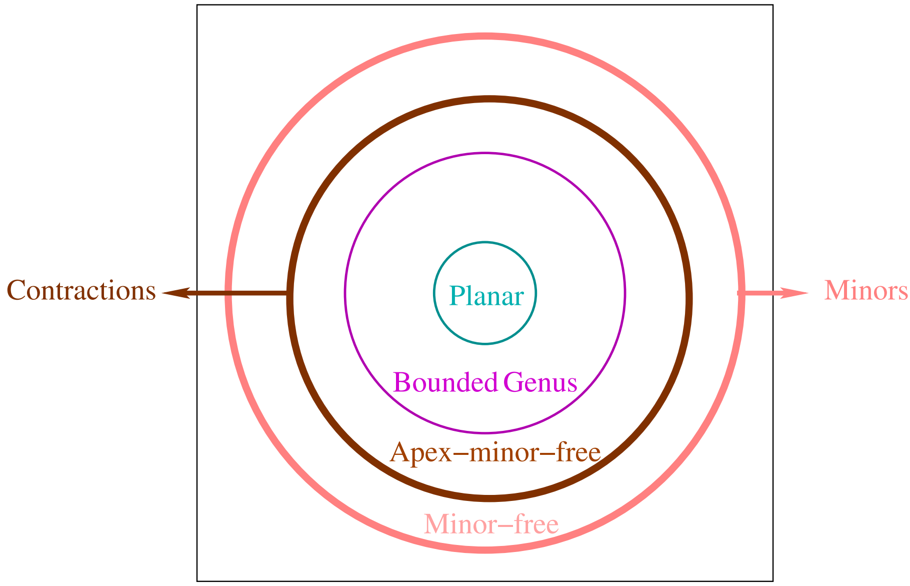


Figure 8.3: Bidimensionality Theory on minors and contractions.

that vertex cover is bidimensional under taking of minors implies the following. If the tree-width of a graph G is $\Omega(\sqrt{k})$ then the graph contains the grid of size $(2\sqrt{k} \times 2\sqrt{k})$, which does not admit a vertex cover of size k. Then, we may immediately answer that G does not admit a vertex cover of size k. Briefly, for an subexponential algorithm for p-VERTEX COVER, where the input is a graph G that does not contain some graph H as a minor, we follow the steps below.

1. We run Amir's constant approximation algorithm for tree-width. [12]
2. If $\mathbf{tw}(G) = \Omega(f(\mathbf{n}(H)))\sqrt{k}$ then the algorithm returns NO, otherwise $\mathbf{tw}(G) = O(f(\mathbf{n}(H)))\sqrt{k}$.
3. We run the $2^{O(\mathbf{tw}(G))}n^{O(1)}$ time algorithm for p-VERTEX COVER.

As $\mathbf{tw}(G) = O(f(\mathbf{n}(H)))\sqrt{k}$ we conclude that p-Vertex Cover can be solved in time $2^{O(\sqrt{k})}n^{O(1)}$.

8.3.2 Bidimensionality Theory for (Distance) Topological Minors

Although up to now bidimensionality of a parameter has only been defined for graph parameters that are closed under taking of minors or contractions, as above, we may define the notion of bidimensionality for graph parameters that are closed under taking of topological minors or distance topological minors.

A parameter P is called *bidimensional under taking of topological minors with density* δ if

1. P is closed under taking of topological minors and
2. If W_k is the wall of height k then $P(W_k) = (\delta k)^2 + o((\delta k)^2)$.

Similarly, a parameter P is called *bidimensional under taking of distance topological minors with density* δ if

1. P is closed under taking of distance topological minors and
2. If W_k is the wall of height k then $P(W_k) = (\delta k)^2 + o((\delta k)^2)$.

As above, if there exists an FPT algorithm that computes the parameter in single exponential time on tree-width, then Theorems 6.4 and 8.3 ensure the existence of a subexponential algorithm for that paremeter in the case where the input is a graph that excludes some graph H as a minor. Note here that in the case where the parameter is closed under taking of distance topological minors then the excluded graph H should be an apex graph. (See Figure 8.4.)

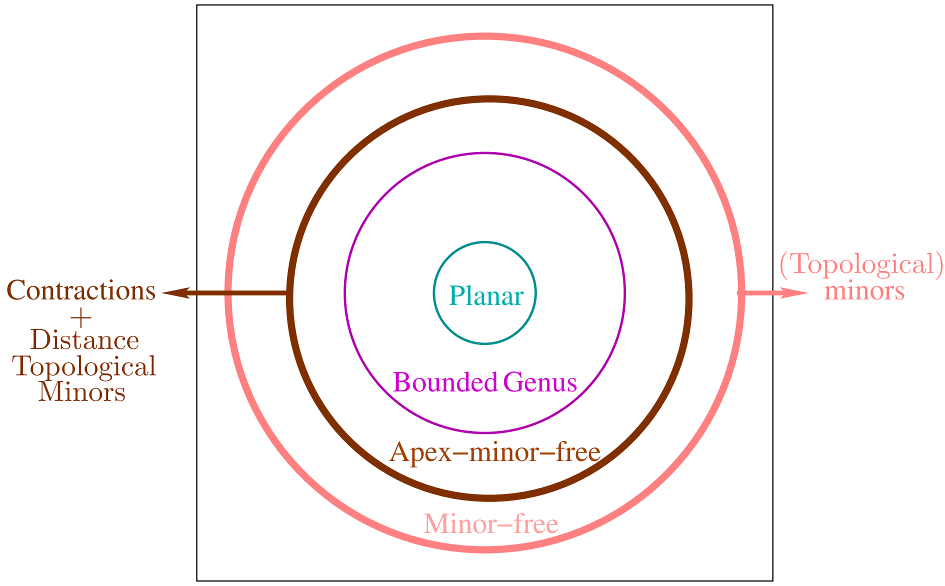


Figure 8.4: The complete Bidimensionality Theory.

8.4 An Application to Cycle Domination and Scattered Cycle Set

8.4.1 Introduction to the Erdős-Pósa Property

In 1965, P. Erdős and L. Pósa proved that there is a function $f : \mathbb{N} \to \mathbb{R}$ such that, given any $k \in \mathbb{N}$, every graph contains either k vertex-disjoint cycles or a set of at most

$$f(k) = \begin{cases} 4k \log k + 4k \log\log k + 17k - 1 & \text{if } k \geq 2 \\ 1 & \text{if } k = 1 \end{cases} \tag{8.1}$$

vertices meeting all its cycles [76].

More generally, let $\mathcal{H}$ be a (not necessarily finite) family of graphs. A *packing* of $\mathcal{H}$ in G is a set of vertex-disjoint subgraphs of G isomorphic to a graph in $\mathcal{H}$. The size of the packing is equal to the number of the vertex-disjoint subgraphs. The *packing number* of $\mathcal{H}$ in G, denoted by $\nu_{\mathcal{H}}(G)$, is equal to the maximum size of a packing in G. The dual notion of packing in graphs is the notion of covering. A *covering* of $\mathcal{H}$ in G is a set of vertices S in G such that the graph $G \setminus S$ does not contain any subgraph isomorphic to a graph in $\mathcal{H}$. The size of a covering S is equal to $|S|$. The *covering number* of $\mathcal{H}$ in G, denoted by $\tau_{\mathcal{H}}(G)$, is equal to the minimum size of a covering in G.

It is easy to observe the following.

Observation 8.1. *Let $\mathcal{H}$ be a family of graphs and G be a graph. If $\nu_{\mathcal{H}}(G) \geq k$ then $\tau_{\mathcal{H}}(G) \geq k$.*

Let $\mathcal{H}$ be the family of all cycles. (In this case the set S for which $|S| = \tau_{\mathcal{H}}(G)$ is also called *feedback vertex set* of G.) Then the Erdős-Pósa theorem can be restated as below.

Theorem 8.5. *Let G be a graph and $k \in \mathbb{N}$. If $\nu_{\mathcal{H}}(G) < k$ then $\tau_{\mathcal{H}}(G) < f(k)$, where f is the function 8.1 and $\mathcal{H}$ is the family of all cycles.*

Erdős-Pósa property of a graph class $\mathcal{H}$. We say that a family of graphs $\mathcal{H}$ has the *Erdős-Pósa property* if there exists a function $f : \mathbb{N} \to \mathbb{R}$ such that

$$\nu_{\mathcal{H}}(G) \leq \tau_{\mathcal{H}}(G) \leq f(\nu_{\mathcal{H}}(G)).$$

It was shown by R. Diestel in [60] that if $\mathcal{H}$ is the class of all graphs that can be contracted to a fixed planar graph H, then $\mathcal{H}$ has the Erdős-Pósa property for an exponential function f. The bound on f was later made linear by F. Fomin, S. Saurabh and D. M. Thilikos for the case where G belongs to a fixed non-trivial minor-closed graph class $\mathcal{G}$ [91].

In the case where $\mathcal{H}$ is the family consisting of all directed cycles and G is a directed graph the Erdős-Pósa property was conjectured by T. Gallai [74] in 1968 and for all directed graphs by D. Younger in 1973 [222]. The case where $k = 2$, was proved in [155] by W. McCuaig. The planar case was resolved by B. Reed and F. Shepherd in [183]. Finally, the general case was resolved by B. Reed, N. Robertson, P. Seymour, and R. Thomas [182]. Specifically,

Theorem 8.6 ([182]). *There exists a function $f : \mathbb{N} \to \mathbb{R}$ such that, given a directed graph G and $k \in \mathbb{N}$, G has either k vertex-disjoint directed cycles or a set S of vertices, with $|S| \leq f(k)$, that meets all directed cycles.*

In this section we consider the extension of the Erdős-Pósa property in two graph parameters other than the packing number and the covering number. In particular, let $\mathcal{H}$ be a family of graphs and G be a graph. The *r-scattering number* of $\mathcal{H}$ in G, is the maximum number of vertex-disjoint copies of graphs of $\mathcal{H}$ in G whose closed neighborhoods at distance r are mutually disjoint. We also define the *r-dominating number* of $\mathcal{H}$ in G, as the minimum size of a vertex set S in G such that for every copy of a graph $H \in \mathcal{H}$, $S \cap N_G^r[V(H)] \neq \emptyset$.

Very recently, Z. Dvořák proved that if $\mathcal{H}$ consists of the graph with one vertex, then the r-scattering number and the r-dominating number have the Erdős-Pósa property [68] for all graphs $G \in \mathcal{F}$, where $\mathcal{F}$ is a graph class of bounded expansion.

In the next subsection, inspired by a proof in [91], we prove that if $\mathcal{H}$ is the family of all cycles and $r = 1$, then the 1-scattered number of $\mathcal{H}$ in G and the 1-dominating number of $\mathcal{H}$ in G have the Erdős-Pósa property when G is a graph that excludes a fixed apex graph as a minor.

8.4.2 The Proof

Let $\mathcal{F}_c$ be the family of all cycles. We define the *cycle domination number of* G as

$$\begin{aligned}\mathbf{cdom}(G) \quad = \quad & \min\{k \mid \exists D \subseteq V(G) : [(|D| \leq k) \text{ and} \\ & (\forall G' \subseteq G)(G' \in \mathcal{F}_c \implies \exists v \in D(N(v) \cap V(G') \neq \emptyset))\},\end{aligned}$$

that is, $\mathbf{cdom}(G) \leq k$ if there exists a set D of cardinality at most k such that for every cycle C in G there is a vertex $v \in D$ with $\mathbf{dist}(v, C) \leq 1$. (See, for example, Figure 8.5.)

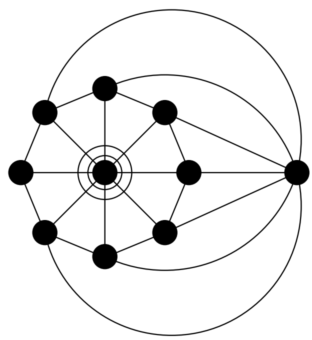


Figure 8.5: A graph with cycle domination number 1.

For every graph G, the *scattered cycle number of* G is defined as

$$\begin{aligned}\mathbf{scs}(G) \quad = \quad & \max\{k \mid \exists V_1, V_2, \ldots, V_k \text{ subsets of } V(G) \text{ such that} \\ & \forall i, j \in [k] \text{ with } i \neq j, N[V_i] \cap N[V_j] = \emptyset \text{ and} \\ & \forall_{i \in [k]} \exists_{C \in \mathcal{F}_c} : C \subseteq G[V_i]\}.\end{aligned}$$

In other words, $\mathbf{scs}(G) \geq k$ if G contains k cycles $C_1, C_2, \ldots, C_k$ whose neighborhoods are disjoint. (See, for example, Figure 8.6.)

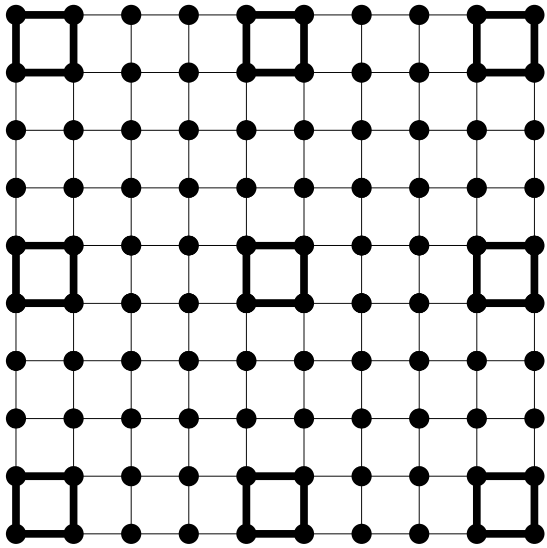


Figure 8.6: A graph with scattered cycle number 9.

Observation 8.2. *For every graph G,* $\mathbf{scs}(G) \leq \mathbf{cdom}(G)$*.*

Observation 8.3. *If H and G are graphs such that H is a contraction of G then* $\mathbf{scs}(H) \leq \mathbf{scs}(G)$*.*

In this section we prove that the cycle domination number and the scattered cycle number satisfy the Erdős-Pósa property in the classes of graphs that exclude a fixed apex graph as a minor.

Theorem 8.7. *Let $\mathcal{G}$ be a graph class that excludes a fixed apex graph as a minor. There is a constant $c_{\mathcal{G}}$ (depending only on $\mathcal{G}$) such that for every graph $G \in \mathcal{G}$, it holds that*

$$\mathbf{scs}(G) \leq \mathbf{cdom}(G) \leq c_{\mathcal{G}} \cdot \mathbf{scs}(G).$$

Recall here that Theorem 6.3 asserts the existence of the triangulated grid Γ_k as a contraction in a connected graph of large enough tree-width that excludes a fixed graph H as a minor.

By applying Theorem 6.3 we obtain the following simple lemma that is crucial in the proof of Theorem 8.7.

Lemma 8.3. *If $\mathcal{G}$ be a graph class that excludes a fixed apex graph H as a minor then there is a constant $\sigma_{\mathcal{G}}$ depending only on $\mathcal{G}$ such that for any graph $G \in \mathcal{G}$, $\mathbf{tw}(G) \leq \sigma_{\mathcal{G}} \cdot \mathbf{scs}(G)$.*

Proof. Let $\mathbf{scs}(G) \leq k$. If $m = \lceil k^{1/2} \rceil + 1$, then

$$\mathbf{scs}(\Gamma_{4m}) > k.$$

As Γ_{4m} contains a scattered cycle set of size k, Observation 8.3 implies that G does not contain Γ_{4m} as a contraction.

Then, from Theorem 6.3, there is a constant $\sigma_{\mathcal{G}}$ depending only on $\mathcal{G}$ such that $\mathbf{tw}(G) \leq \sigma_{\mathcal{G}} \cdot m$ and this concludes the proof of the lemma. (See, for example, Figure 8.7) □

For our purposes, we enhance the definition of a tree decomposition $(\mathcal{X}, T)$ as follows; T is a rooted tree on some node r where $X_r = \emptyset$ and each one of its nodes have at most two children that are one of the kinds below.

1. *Introduce node*: A node t that has only one child t' where $X_t \supset X_{t'}$ and $|X_t| = |X_{t'}| + 1$.

2. *Forget node*: A node t that has only one child t' where $X_t \subset X_{t'}$ and $|X_t| = |X_{t'}| - 1$.

3. *Join node*: A node t that has exactly two children t_1 and t_2 such that $X_t = X_{t_1} = X_{t_2}$.

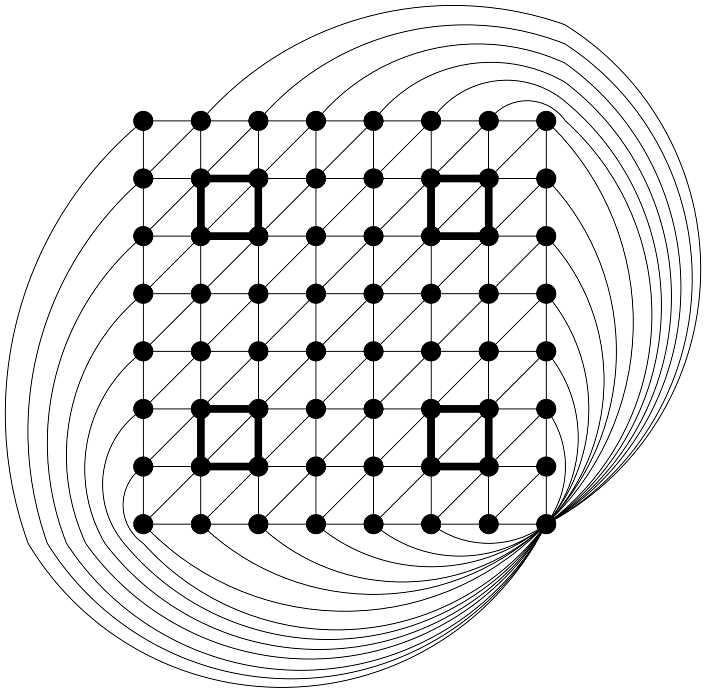


Figure 8.7: A scattered cycle set of size 4 in Γ_8.

4. *Base node*: A node t that is a leaf of T, is different from the root, and $X_t = \emptyset$.

A tree decomposition satisfying the aforementioned properties is called a *nice* tree decomposition. It is easy to see that each tree decomposition can be transformed to a nice tree decomposition while maintaining the same width, for example see [26]). In this section, when we refer to a tree decomposition $(\mathcal{X}, T)$ we presume that it is nice.

Given a tree-decomposition $(\mathcal{X}, T)$ and some node t of T we define as T_t the subtree rooted on t. Clearly, it r is the root of tree, $T_r = T$. We also define $G_t = G[\bigcup_{s \in V(T_t)} X_s]$. For every $t \in V(T)$, we denote its parent in T by $\pi(t)$ and its child by $\alpha(t)$ (in the case where t is a join node we choose arbitrarily one of its chidren).

Evenmore, we define the set $\mathbf{crt}(G, T)$, which we call the set of *critical nodes* of T, in the following recursive way; if t is a leaf then $t \notin \mathbf{crt}(G, T)$, that is, t not a critical node of T. Otherwise, t is critical if and only if $X_t \cup \bigcup_{s \in \mathbf{crt}_{(G_t, T)}} X_{\alpha(s)}$ does not dominate all cycles of G_t but $X_{\alpha(t)} \cup \bigcup_{s \in \mathbf{crt}_{(G_t, T)}} X_{\alpha(s)}$ does. Notice that if $t \in V(T)$ is a critical node, then t is also a forget node. For any graph G, let

$$\mathbf{crt}(G) = \min\{\mathbf{crt}(G, \mathcal{T}) \mid \mathcal{T} \text{ is a tree decomposition of } G\}.$$

Observation 8.4. *Let k be a non-negative integer. If G is a graph such that $\mathbf{crt}(G) > k$ then $\mathbf{scs}(G) > k$.*

Given a graph G, we call a triple (V_1, S, V_2) *d-separation triple* of G if $|S| \leq d$ and $\{V_1, S, V_2\}$ is a partition of $V(G)$ such that there is no edge in G with a vertex in V_1 and a vertex in V_2.

Lemma 8.4. *Let $\mathcal{G}$ be a class of graphs that exclude a fixed apex graph H as a minor and let $G \in \mathcal{G}$ such that $\mathbf{scs}(G) = k \geq 1$. Then there is a $\sigma_{\mathcal{G}} \cdot \sqrt{k}$-separation triple (V_1, X, V_2) of G, with $\frac{k}{3} \leq \mathbf{crt}(G[V_1]) \leq 2\frac{k}{3}$ and $\mathbf{crt}(G[V_1]) + \mathbf{crt}(G[V_2]) \leq k$, where $\sigma_{\mathcal{G}}$ is a constant that depends only on $\mathcal{G}$.*

Proof. From Lemma 8.3, there is a tree decomposition $\mathcal{T} = (\mathcal{X}, T)$ of G of width at most $\sigma_{\mathcal{G}}\sqrt{k}$, where $\sigma_{\mathcal{G}}$ is a constant that depends only on $\mathcal{G}$. Let $q : V(T) \to \mathbb{N}$ such that $q(t) = \mathbf{crt}(G_t, T)$. Observe that:

(i.) If t is a leaf, then $q(t) = 0$ from the definition.

(ii.) If t is an introduce node then $q(t) = q(\alpha(t))$ as $X_t \subseteq X_{\alpha(t)}$.

(iii.) If t is a forget node, $q(t) - q(\alpha(t)) \in \{0, 1\}$. This holds because t can be either a critical node or not.

(iv.) If t is a join node, then $q(t) = q(t_1) + q(t_2)$, where t_1 and t_2 are its children in T as $q(t)$ cannot be a critical node.

(v.) $q(r) = \mathbf{crt}(G, \mathcal{T})$ if r is the root, as $G_r = G$.

From the above follows that there exists a unique node $t \in V(T)$ such that $q(t) > 2\frac{k}{3}$ and if t' is a child of t, then $q(t') \leq 2\frac{k}{3}$. Notice that t is either a forget or a join node.

In the case where t is a forget node, let $V_1 = V(G_t) \setminus X_t$, $V_2 = V(G) \setminus (V_1 \cup X_t)$, and $X = X_t$, and observe that $\mathbf{crt}(G[V_1]) \leq \left\lfloor 2\dfrac{k}{3} \right\rfloor$ and $\mathbf{crt}(G[V_2]) \leq \left\lfloor 2\dfrac{k}{3} \right\rfloor$.

In the case where t is a join node, then $q(t_1) \leq 2\frac{k}{3}$ and $q(t_2) \leq 2\frac{k}{3}$ but $q(t_1) + q(t_2) = q(t) > 2\frac{k}{3}$, where t_1 and t_2 are the children of t. This implies that, without loss of generality, $q(t_1) \leq \frac{k}{3}$. Let $V_1 = V(G_{t_1}) \setminus X_{t_1}$, $V_2 = V(G) \setminus (V_1 \cup X_{t_1})$, and $X = X_{t_1}$. We conclude that in both cases $\frac{k}{3} \leq \mathbf{crt}(G[V_1]) \leq 2\frac{k}{3}$ and $\mathbf{crt}(G[V_1]) + \mathbf{crt}(G[V_2]) \leq k$. □

We our now able to prove our main theorem.

Proof of Theorem 8.7. Let G be a graph in $\mathcal{G}$. Notice that it is enough to prove the second inequality. Using induction on $\mathbf{scs}(G)$ we will prove that

$$\mathbf{cdom}(G) \leq \beta \cdot \sigma_{\mathcal{G}} \cdot \mathbf{crt}(G) - \gamma \cdot \sigma_{\mathcal{G}} \cdot \sqrt{\mathbf{crt}(G)}.$$

When $\mathbf{scs}(G) = 0$, the claim holds trivially. Assume now that $\mathbf{scs}(G) = k$, where $k \geq 1$. From Lemma 8.4, G contains a $\sigma_{\mathcal{G}} \cdot \sqrt{k}$-separation triple, where $\frac{k}{3} \leq \mathbf{crt}(G[V_1]) \leq 2\frac{k}{3}$ and $\mathbf{crt}(G[V_1]) + \mathbf{crt}(G[V_2]) \leq k$. Observe that $\mathbf{cdom}(G) \leq \mathbf{cdom}(G_1) + \mathbf{cdom}(G_2) + |X|$.

The induction hypothesis implies that for some $\delta \in [\frac{1}{3}, \frac{2}{3}]$,

$$\begin{aligned}\mathbf{cdom}(G) \ \leq \ & \beta \cdot \sigma_{\mathcal{G}} \cdot \delta \cdot k - \gamma \cdot \sigma_{\mathcal{G}} \sqrt{\delta \cdot k} + \\ & \beta \cdot \sigma_{\mathcal{G}} \cdot (1-\delta) \cdot k - \gamma \cdot \sigma_{\mathcal{G}} \sqrt{(1-\delta) \cdot k} + \sigma_{\mathcal{G}} \cdot \sqrt{k}.\end{aligned}$$

This is upper bounded by $\beta = 3.54$ and $\gamma = 2.54$. Thus, the theorem follows from Observation 8.4 for $c_{\mathcal{G}} = 3.54 \cdot \sigma_{\mathcal{G}}$. □

8.5 Kernelization

In this section we present another notion from Parameterized Complexity that is going to be useful in the next chapter, the notion of kernelization.

In kernelization our aim is to preprocess an instance of a hard problem so that we obtain a new equivalent instance of smaller size. Then, as the instance of our problem has shrunk a lot we may apply a "brute force" algorithm to solve it. Notice here that we want the preprocessing to run efficiently (in particular, in polynomial time).

Formally, a parameterized problem Π is said to admit a $g(k)$ *kernel* if there is a polynomial time algorithm that transforms any instance (x, k) to an equivalent instance (x', k') such that $|x'| \leq g(k)$ and $k' \leq g(k)$. If $g(k) = k^{O(1)}$ or $g(k) = O(k)$ we say that Π admits a *polynomial kernel* and *linear kernel* respectively.

The *kernelization* of a parameterized occurs with the assistance of reduction rules. A *reduction rule* is a polynomial time algorithm that takes an an input an instance (I, k) of a problem Π and outputs an equivalent instance (I', k') of Π. We apply these reduction rules so to obtain a kernel of Π.

The notion of kernelization is really important in Parameterized Complexity Theory as it equivalent to the notion of fixed parameter tractability.

Theorem 8.8 ([171]). *For every parameterized problem* (Π, κ)*, the following are equivalent:*

1. (Π, κ) *is fixed parameter tractable.*

2. Π *is decidable and* (Π, κ) *admits a kernel.*

We would like to mention here that many times this theorem appears erroneously without the condition that Π is decidable.

For example, let's see how we may obtain a quadratic kernel for k-Vertex Cover.

Observation 8.5. *Let* G *be a graph for which we try to find a vertex cover of size at most* k*. If there exists a vertex* $v \in V(G)$ *such that* $\deg_G(v) > k$*, then* v *belongs to all vertex covers of* G *of size at most* k*.*

Indeed, notice that if v is not contained in a vertex cover V of G of size at most k, then all its neighbors belong to V. This however is a contradiction since v has more than k neighbors.

Observation 8.6. *If* G *is a graph with* $\Delta(G) \leq k$ *that admits a vertex cover of size* k*, then* G *has at most* k^2 *edges.*

Then the following is a quadratic kernel for k-Vertex Cover.

First we check whether $k = 0$. In this case if G has no edges we output a trivially positive instance of the problem otherwise we output a trivially negative instance.

Otherwise, if $k > 0$ then if G has a vertex v with $\deg_G(v) > k$, we recursively run the algorithm with input $(G - v, k - 1)$. Otherwise, we check whether G has at most k^2 edges. If yes, we output (G, k), otherwise we output a trivially negative instance.

We would like to mention here that the above kernelization is attributed to Buss and is called *Buss Kernelization* [66].

This is a simple example of a kernelization technique. There are many more techniques either for single problems (see, for example [9, 47, 90]) or for classes of problems (see, for example [24, 135]). However, further mentioning these techniques is out of the purposes of this section.

Finally, we would like to mention that there exist many proofs of expected lower bounds on the size of the kernel of a fixed parameter tractable problem. The lower bound occurs under the condition that if a kernel of smaller size is existent then some unresolved conjecture of Parameterized Complexity fails. For example, in Classical Complexity, it is known that 3-SAT is not solvable in polynomial time unless $\mathsf{P} = \mathsf{NP}$. The most famous of these conditions when we work on lower bounds of kernels is the *Exponential Time Hypothesis* that was proposed by R. Impagliazzo, R. Paturi, and R. Zane [126].

Definition 8.1 (Exponential Time Hypothesis)**.** Let ϕ be an instance of 3-SAT on n variables and m clauses. There does not exists an $2^{o(m)}$ algorithm that decides whether there exists a satisfying assignment on the variables of ϕ.

In the next chapter we study a parameterization of SET SPLITTING.

CHAPTER 9

Algorithms and Kernels on General Graphs

9.1 Introduction to the Bipartizations of (Hyper)graphs

Max Cut is a well known classical problem. Here, the input is a graph G and a positive integer k and the objective is to check whether there is a cut of size at least k. A *cut* of a graph is a *bipartition* of the vertices of a graph into two disjoint subsets. See, for example, the bipartition that is induced by the vertex sets of same color in Figure 9.1.

The *size* of the *cut* is the number of edges whose endpoints are in different subsets of the bipartition. Max Cut is NP-hard and has been the focus of extensive study, from the algorithmic perspective in Computer Science as well as the extremal perspective in Combinatorics. In this chapter we focus on a generalization of Max Cut to hypergraphs and study this

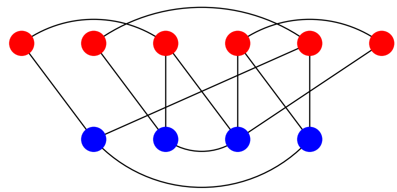


Figure 9.1: A cut of a graph.

generalization with respect to Extremal Combinatorics and Parameterized Complexity.

Recall that a hypergraph H consists of a vertex set $V(H)$ and a set $E(H)$ of subsets of $\mathcal{P}(V(H))$, called hyperedges. A *hyperedge* $e \in E(H)$ is a subset of the vertex set $V(H)$. By $V(e)$ we denote the subset of vertices corresponding to the hyperedge e. A hypergraph is called an r*-hypergraph* if the size of each hyperedge is upper bounded by r. Given a hypergraph 2-coloring, $\phi : V(H) \to \{-1, 1\}$, we say that it splits a hyperedge e if $V(e)$ has a vertex assigned 1 as well as a vertex assigned -1 under ϕ. In Max r-Set Splitting, a generalization of Max Cut, we are given a hypergraph H and a positive integer k, and the objective is to check whether there exists a coloring function $\phi : V(H) \to \{-1, 1\}$ such that at least k hyperedges are split. This problem is the main topic of this chapter.

For a graph G, let $\zeta(G)$ be the size of a maximum cut. Erdős [75] observed that $\zeta(G) \geq m/2$ for graphs with m edges. To see this notice that a random bipartition of the vertices of a graph G with m edges gives a cut with size at least $m/2$. A natural question was whether the bound on ζ could be improved. Answering a question of Erdős [75], Edwards [70] proved that for any graph G on m edges $\zeta(G) \geq \left\lceil \frac{m}{2} + \sqrt{\frac{m}{8} + \frac{1}{64}} - \frac{1}{16} \right\rceil$. In the same paper Edwards also showed that for every connected graph

G on n vertices and m edges, $\zeta(G) \geq \frac{m}{2} + \frac{n-1}{4}$. These bounds are known to be tight (see [27] for a survey on this area). The first result of this chapter generalizes this classical result. For an r-hypergraph H, let $\zeta(H)$ be the maximum number of hyperedges that can be split by a hypergraph 2-coloring. Let H be a hypergraph with vertex set $V(H)$, and edge set $E(H) = \{e_1, e_2, \ldots e_m\}$. Observe that a random 2-coloring that sets each vertex of hypergraph H to 1 or -1 with equal probability always splits at least $\mu_H = \sum_{i=1}^{m}(1 - 2/2^{|e_i|}) = \sum_{i=1}^{m}(1 - 2^{1-|e_i|})$ number of hyperedges. Indeed, to see this, notice that there exist $2^{|e_i|}$ possible 2-colorings of its vertices and only two of them do not split e_i (the ones where all vertices are assigned color either 1 or -1). Then the linearity of expectation yields the above equality.

We show that if an r-hypergraph H is "partition connected" then $\zeta(H) \geq \mu_H + \frac{n-1}{r2^{r-1}}$.

Theorem 9.1. *Let H be a partition connected r-hypergraph with an n sized vertex set $V(H)$ and edge set $E(H) = \{e_1, e_2, \ldots, e_m\}$. Then $\zeta(H) \geq \mu_H + \frac{n-1}{r2^{r-1}}$, where $\mu_H = \sum_{i=1}^{m}(1 - 2^{1-|e_i|})$.*

Since the definition of partition connectivity coincides with the definition of connectivity on graphs, for partition connected uniform 2-hypergraphs (every hyperedge has size exactly 2), $\zeta(H) \geq \frac{m}{2} + \frac{n-1}{4}$. The notion of uniform 2-hypergraphs is same as that of ordinary graphs, thus, for $r = 2$, we get the old result of Edwards. Proof of Theorem 9.1 could also be thought of as a generalization of a similar proof obtained in [38] for ordinary graphs.

We use our combinatorial result to study an above guarantee version of Max r-Set Splitting in the realm of parameterized complexity. Studies on problems parameterized above guaranteed combinatorial bounds are in vogue. A simple example of such a problem is the decision problem that takes as input a planar graph on n vertices and an integer k, where the parameter is k, and asks if there is an independent set of size at least $\frac{n}{4} + k$. An independent set of size at least $n/4$ is guaranteed by the Four Color Theorem. Could this problem be solved in time $O(n^{g(k)})$, for some function g? Is there an FPT algorithm? No one knows. This is a nice and simple example of this research theme, which is quite well-motivated and that has developed strongly since it was introduced by Mahajan and Raman [151].

Mahajan and Raman showed that several above guarantee versions of Max Cut and Max Sat are FPT. Later, Mahajan et al. [152] published a paper with several new results and open problems around parameterizations beyond guaranteed lower and upper bounds. In a breakthrough paper Gutin et al. [112] developed a probabilistic approach to problems parameterized above or below tight bounds. Alon et al. [8] combined this approach with methods from Algebraic Combinatorics and Fourier Analysis to obtain an FPT algorithm for parameterized Max r-Sat beyond the guaranteed lower bound. Other significant results in this direction include quadratic kernels for ternary permutation constraint satisfaction problems parameterized above average and results around systems of linear equations over field with two elements [38, 39, 113, 136].

A standard parameterized version of Max r-Set Splitting is defined by asking whether there exists a hypergraph 2-coloring that splits at least k hyperedges. This version of Max r-Set Splitting, called p-Set Splitting, has been extensively studied in parameterized algorithms. In p-Set Splitting we *do not* restrict the size of hyperedges to at most r as in the case of Max r-Set Splitting. Dehne, Fellows, and Rosamond [46] initiated the study of p-Set Splitting and gave an algorithm running in time $O^*(72^k)$. They also provided a kernel for the problem with at most

History of p-Set Splitting		
Dehne, Fellows, and Rosamond	WG 2003	$O^*(72^k)$
Dehne, Fellows, Rosamond, and Shaw	IWPEC 2004	$O^*(8^k)$
Lokshtanov and Sloper	ACiD 2005	$O^*(2.6499^k)$
Chen and Lu	COCOON 2007	$O^*(2^k)$
Lokshtanov and Saurabh	IWPEC 2009	$O^*(1.96^k)$
Nederlof and van Rooij	IPEC 2010	$O^*(1.8213^k)$

Table 9.1: List of known results about p-Set Splitting in chronological order. The $O^*()$ notation suppresses the polynomial factor.

$2k$ hyperedges. Later Dehne, Fellows, Rossmand, and Shaw [47] obtained an algorithm with running time $O^*(8^k)$. Continuing this chain of improvement Lokshtanov and Sloper [150] gave an algorithm with running time $O^*(2.65^k)$ and obtained a kernel with both the number of vertices and the number of hyperedges at most $2k$. Later, Chen and Lu [33] provided a randomized algorithm with running time $O^*(2^k)$ for a weighted version of problem. In 2009, Lokshtanov and Saurabh [149] gave an algorithm with running time $O^*(1.96^k)$ and a kernel with at most $2k$ hyperedges and at most k variables. The current fastest algorithm is given by Nederlof and van Rooij [159] and runs in time $O^*(1.8213^k)$. We refer to Table 9.1 for a quick reference on the history of the p-Set Splitting problem.

From now onwards we only consider r-hypergraphs. If we have a hyperedge of size one then it can never be split and hence we can remove it from consideration. So we assume that every hyperedge is of size at least 2 and at most r. Let H be a hypergraph with vertex set $V(H)$ and edge set $E(H) = \{e_1, e_2, \dots e_m\}$. Since every hyperedge is of size at least 2, we have that $\mu_H \geq m/2$. Thus, the standard parameterization of Max r-Set

Splitting is trivially FPT because of the following argument. If $k \leq m/2$ then the answer is yes else we have that $m \leq 2k$ and hence $n \leq 2kr$. In this case we can enumerate all the $\{1, -1\}$-colorings to $V(H)$ and check whether anyone of them splits at least k hyperedges and answer accordingly. Thus given an r-hypergraph H, the more meaningful question is whether there exists a $\{1, -1\}$-coloring of $V(H)$ that splits at least $\mu_H + k$ clauses. In other words, we are interested in the following above average version of Max r-Set Splitting.

Above Average r-Set Splitting (AA-r-SS)	
Instance:	An r-hypergraph H and a non-negative integer k.
Parameter:	k.
Question:	Does there exist 2-coloring of $V(H)$ that splits at least $\mu_H + k$ hyperedges?

It is known by the results in [136] that AA-r-SS is FPT for a constant r ($r = O(1)$). From an algorithmic point of view, a natural question is whether AA-r-SS is FPT when the sizes of hyperedges is at most $r(n)$ for some function of n. If yes, how far can we push the function $r(n)$? On the algorithmic side, using Theorem 9.1 we get the following result.

Theorem 9.2. *For every fixed constant $\alpha < 1$,* AA-$\alpha \log n$-SS *is FPT.*

We complement the algorithmic result by a matching lower bound result which states the following.

Theorem 9.3. *Unless* $\mathsf{NP} \subseteq \mathrm{DTIME}(n^{\log \log n})$, AA-$\lceil \log n \rceil$-SS *is not in XP.*

Theorems 9.2 and 9.3 are in sharp contrast to a similar question about AA-Max-r-Sat. Let F be a CNF formula on n variables and m clauses and let $r_1, \ldots, r_m$ be the number of literals in the clauses of F. Then

$\mathrm{asat}(F) = \sum_{i=1}^{m}(1 - 2^{-r_i})$ is the expected number of clauses satisfied by a random truth assignment (the truth values to the variables are distributed uniformly and independently). In AA-MAX-r-SAT we are given a r-CNF formula F (all clauses are of size at most r) and a positive integer k, and the question is whether there is an assignment that satisfies at least $\mathrm{asat}(F)+k$ clauses. Here k is the parameter. In [40], it is shown that AA-MAX-$r(n)$-SAT is not FPT unless Exponential Time Hypothesis fails [126], where $r(n) \geq \log\log n+\phi(n)$ and $\phi(n)$ is any unbounded strictly increasing function. However, they also show that MAX-$r(n)$-SAT-AA is FPT for any $r(n) \leq \log\log n - \log\log\log n - \phi(n)$, where $\phi(n)$ is any unbounded strictly increasing function.

The proof of Theorem 9.2 also shows that AA-r-SS admits a kernel with $O(k)$ vertices for fixed r. Earlier, only a linear "bikernel" was known [136]. The proofs of Theorem 9.1 and 9.2 combine the properties of Fourier coefficients of pseudo-Boolean functions, observed by Crowston et al. [38], with results on a certain kind of connectivity of hypergraphs. The proof of Theorem 9.3 is inspired by a similar proof given in [40].

9.2 New Lower Bound on $\zeta(H)$ and Proof of Theorem 9.1

In this section we obtain the new lower bound on $\zeta(H)$, the maximum number of hyperedges that can be split in an r-hypergraph H by a hypergraph 2-coloring. Towards this we first define the notion of hypergraph connectivity and hypergraph spanning tree.

Hypergraph Connectivity and Hypergraph Spanning Tree. With every hypergraph H we can associate the following graph: The *primal graph*, also called the *Gaifman graph*, $P(H)$ has the same vertices $V(H)$ as H and two vertices $u, v \in V(H)$ are connected by an edge in $P(H)$ if there is a hyperedge $e \in E(H)$, such that $\{u, v\} \subseteq V(e)$. We say that H is *connected* or has r components if the corresponding primal graph $P(H)$ is connected or has r components. Now we define the notions of strong cut-sets and forests in hypergraphs.

Definition 9.1 (Strong Cut-Set and Partition Connectivity)**.** A subset $X \subseteq E(H)$ is called a *strong cut-set* if the hypergraph $H' = (V, E(H) \setminus X)$ has at least $|X| + 2$ connected components. A hypergraph H is *partition connected* if it does not have a strong cut-set.

Definition 9.2 (Hypergraph Forest)**.** A *forest* $\mathcal{F}$ of a hypergraph H is a pair (F, g) where F is a forest, in the normal graph theoretic sense, with vertex set $V(H)$ and edge set $E(F)$, and $g : E(F) \to E(H)$ is an injective map such that for every $\{u, v\} \in E(F)$ we have $\{u, v\} \subseteq V(g(\{u, v\}))$. The number of edges in $\mathcal{F}$ is $|E(F)|$.

Observe that if a forest $\mathcal{F}$ has $|V(H)| - 1$ edges then F is a spanning tree on $V(H)$. In this case we say that $\mathcal{F}$ is a *hypertree* of H. Frank, Király, and Kriesell proved the following duality result relating spanning trees and strong cut-set in hypergraphs [93, Corollary 2.6].

Proposition 9.1 ([93])**.** *A hypergraph H contains a hypertree if and only if H does not have a strong cut-set.*

A 2-*coloring* of a hypergraph H is a function $c : V(H) \to \{-1, 1\}$. We say that a hyperedge e of H is *split* by c if some vertex in $V(e)$ is assigned 1 and some vertex is assigned -1. We denote by $\mathbf{split}(c, H)$ the set of hyperedges split by c. (We may omit the hypergraph if it is clear from the context.) The maximum number of hyperedges split over all such 2-colorings is denoted by $\mathbf{split}(H)$.

Observation 9.1. *Let H be a hypergraph, e be a hyperedge of H, and $v \in V(e)$ be a vertex of H. If c is a 2-coloring of H then e is not split if and only if $c(v) \cdot c(u) = 1$ for every $u \in V(e) \setminus \{v\}$.*

For every $i \geq 2$, let m_i be the number of hyperedges of H that have size i. For every r-hypergraph H, we rewrite μ_H as follows, $\mu_H = \sum_{i=2}^{r}(1 - 2^{-(i-1)})m_i$.

Let H be a hypergraph that does not have a strong cut-set. Here, we will show that for such hypergraphs, there exists a 2-coloring that splits far more than the average. This will be crucial both for our kernelization (Theorem 9.7) and algorithmic (Theorem 9.2) results. For this we will also need a result on boolean functions.

Results from Boolean Functions. A function f that maps $\{-1, 1\}^n$ to $\mathbb{R}$ is called a *pseudo-boolean* function. It is well known that every pseudo-boolean function f can be uniquely written as

$$f(x_1, \ldots, x_n) = \hat{f}(\emptyset) + \sum_{I \in 2^{[n]} \setminus \emptyset} \hat{f}(I) \prod_{i \in I} x_i,$$

where each $\hat{f}(I)$ is a real. This formula is called the *Fourier expansion* of f and the $\hat{f}(I)$ are the Fourier coefficients of f. See [177] for more details. By $\bar{x}$ we represent $(x_1, \ldots, x_n)$.

Theorem 9.4 ([38]). *Let $f(\bar{x}) = \hat{f}(\emptyset) + \sum_{I \in \mathcal{F}} \hat{f}(I) \prod_{i \in I} x_i$ be a pseudo-boolean function of degree $r > 0$, where $\mathcal{F}$ is a family of non-empty subsets of $[n]$ such that $I \in \mathcal{F}$ if and only if $\hat{f}(I) \neq 0$, and $\hat{f}(\emptyset)$ is the constant term of f. Then*

$$\max_{\bar{x} \in \{-1,1\}^n} f(\bar{x}) \geq \hat{f}(\emptyset) + \left\lfloor \frac{rankA - 1 + r}{r} \right\rfloor \cdot \min\{|\hat{f}(I)| \mid I \in \mathcal{F}\},$$

where A is a $(0,1)$-matrix with entries α_{ij} such that $\alpha_{ij} = 1$ if and only if term j of the sum contains x_i.

Now we are ready to give the proof of Theorem 9.1.

Proof of Theorem 9.1. Let H be an r-hypergraph and $1, \ldots, n$ be an arbitrary ordering of vertices in $V(H)$. Let $x_1, \ldots, x_n$ be n variables corresponding to $1, \ldots, n$ respectively. With every hyperedge $e \in E(H)$ we associate a polynomial $f_e(\bar{x})$. For a given $e \in E(H)$, let j be the largest index inside $V(e)$. Then

$$f_e(\bar{x}) = 1 - \frac{1}{2^{|e|-1}} \prod_{i \in V(e) \setminus \{j\}} (1 + x_i x_j).$$

Notice that for every $\delta = (\delta_1, \delta_2, \ldots, \delta_n) \in \{-1, 1\}^n$, we may define a 2-coloring c_δ of $V(H)$ such that $c_\delta(i) = \delta_i$ and, conversely, for every 2-coloring c we may define a vector $\delta_c \in \{-1, 1\}^n$. Observe then that, given a 2-coloring c of H, $f_e(\delta_c) = 1$ if and only if e is split by c. Thus $f_e(\delta_c) = 0$ if and only if e is not split by c. Hence, it is enough to prove that $\max_{\bar{y}} f(\bar{y}) \geq \mu_H + \frac{n-2}{r \cdot 2^{r-1}}$, where $f(\bar{x}) = \sum\limits_{e \in E(H)} f_e(\bar{x})$ is a pseudo-boolean function of degree $r > 0$ and $\bar{y} \in \{-1, 1\}^n$. Next we show that it indeed holds.

Let,

$$\begin{aligned} f(\bar{x}) &= \sum_{e \in E(H)} \left(1 - \frac{1}{2^{|e|-1}} \prod_{i \in V(e) \setminus \{j\}} (1 + x_i x_j) \right) \\ &= \sum_{i=2}^{r} m_i - \sum_{e \in E(H)} \frac{1}{2^{|e|-1}} \prod_{i \in V(e) \setminus \{j\}} (1 + x_i x_j). \end{aligned}$$

Notice, for every $e \in E(H)$, $\frac{1}{2^{|e|-1}} x_p x_j$ and $\frac{1}{2^{|e|-1}} x_j^2 x_p x_q$ appear in the terms of $\prod\limits_{i \in V(e) \setminus \{j\}} (1 + x_i x_j)$ for every $\{p, q\} \subseteq V(e) \setminus \{j\}$. We use this fact later. We rewrite $f(\bar{x})$ as,

$$\begin{aligned} f(\bar{x}) &= \sum_{i=2}^{r} m_i - \sum_{i=2}^{r} \frac{1}{2^{i-1}} m_i + \sum_{I \in \mathcal{F}} c_I \prod_{i \in I} x_i^{\lambda_{(I,i)}} \\ &= \sum_{i=2}^{r} \left(1 - \frac{1}{2^{i-1}}\right) m_i + \sum_{I \in \mathcal{F}} c_I \prod_{i \in I} x_i^{\lambda_{(I,i)}}, \end{aligned}$$

where $\mathcal{F}$ is a family of subsets of $[n]$ such that for each set $I \in \mathcal{F}$,

1. $2 \leq |I| \leq r$,
2. $|c_I| \geq \frac{1}{2^{r-1}}$, and
3. for every $i \in I$, $\lambda_{(I,i)}$ is a positive integer.

Then, as above, for every $e \in E(H)$, $\dfrac{1}{2^{|e|-1}} x_p x_j$ and $\dfrac{1}{2^{|e|-1}} x_j^2 x_p x_q$ appear in $f(\bar{x})$ for every $\{p, q\} \subseteq V(e) \setminus \{j\}$.
Let,

$$f_p(\bar{x}) = \sum_{i=2}^{r} \left(1 - \frac{1}{2^{i-1}}\right) m_i + \sum_{I \in \mathcal{F}} c_I \prod_{i \in I} x_i^{\lambda_{(I,i)} \mod 2}.$$

Clearly $f_p : \{-1, 1\}^n \to \mathbb{R}$ is a pseudo-boolean function. Then, for every $\bar{x} \in \{-1, 1\}^n$, $f(\bar{x}) = f_p(\bar{x})$. Therefore, $\max_{\bar{x}} f(\bar{x}) = \max_{\bar{x}} f_p(\bar{x})$. Notice that $f_p(\bar{x})$ can also be written as

$$f_p(\bar{x}) = \sum_{i=2}^{r} \left(1 - \frac{1}{2^{i-1}}\right) m_i + \sum_{I \in \mathcal{F}'} c'_I \prod_{i \in I} x_i,$$

where $\mathcal{F}'$ is a family of subsets of $[n]$ such that

1. $2 \leq |I| \leq r$ and
2. $|c'_I| \geq \frac{1}{2^{r-1}}$ for every $I \in \mathcal{F}'$.

Then, for every hyperedge $e \in E(H)$, the term $x_p x_q$, $p, q \in V(e)$ with $p \neq q$ appears in $\sum_{I \in \mathcal{F}'} c'_I \prod_{i \in I} x_i$. Before we proceed we rewrite $f_p(\bar{x})$ as

$$f_p(\bar{x}) = \hat{f}(\emptyset) + \sum_{I \in \mathcal{F}'} \hat{f}(I) \prod_{i \in I} x_i,$$

where $\hat{f}(\emptyset) = \mu_H$ is the constant term of f_p and $\hat{f}(I) = c'_I$, for every $I \in \mathcal{F}'$.

Note that $f_p(\bar{x})$ has degree r_p with $2 \leq r_p \leq r$. From Theorem 9.4, it follows that

$$\max_{\bar{x}} f_p(\bar{x}) \geq \hat{f}(\emptyset) + \left\lfloor \frac{\text{rank}A - 1 + r_p}{r_p} \right\rfloor \cdot \min\{|\hat{f}(I)| : I \in \mathcal{F}'\},$$

where A is a $(0,1)$-matrix with entries α_{ij} such that $\alpha_{ij} = 1$ if and only if term $j \in I$ contains x_i. As H does not contain a strong cut-set, H has a hypertree T (Hypothesis and Proposition 9.1). Moreover, recall that for every hyperedge $e \in E(H)$, the term $x_p x_q$, $p, q \in V(e)$ with $p \neq q$ appears in $f_p(\bar{x})$. Thus, the edge-vertex incidence matrix of T is a submatrix of A. It is known that the edge incidence matrix of a connected graph on n vertices has rank at least $n-1$, thus we have that the $\mathbf{rank}T$ is $n-1$. We also know that the rank of a matrix is at least as much as any of its submatrices. This implies that $\mathbf{rank}A \geq n-1$ and

$$\max_{\bar{x}} f_p(\bar{x}) \geq \hat{f}(\emptyset) + \left\lfloor \frac{n-1-1+r}{r} \right\rfloor \cdot \min\{|\hat{f}(I)| \mid I \in \mathcal{F}'\} \geq \mu_H + \frac{n-1}{r \cdot 2^{r-1}}.$$

To see the last inequality let us assume that $n = pr + q$, where $0 \leq q \leq r-1$. Then if $q \geq 2$ we have that $\left\lfloor p + \frac{q+r-2}{r} \right\rfloor \geq p+1$ and this gives the desired result. In other cases we have $q \leq 1$ and that gives us that $\left\lfloor p + \frac{q+r-2}{r} \right\rfloor \geq p \geq \frac{n-1}{r}$. As $\max_{\bar{x}} f(\bar{x}) = \max_{\bar{x}} f_p(\bar{x})$, this completes the proof by applying Theorem 9.4. $\square$

9.3 Reduction Rules for AA-r-SS

In this section, we present two reduction rules for AA-r-SS. These will be of crucial importance towards the proof of Theorem 9.2.

When the hypergraph H is disconnected we can give a simple reduction rule.

Reduction Rule 1 ([149]). : *Let* (H, k) *be an instance of* AA-r-SS *such that* $P(H)$ *has connected components* $P(H)[C_1], \dots, P(H)[C_t]$. *Let* $v_1, \dots, v_t$ *be vertices such that* $v_i \in C_i$. *Construct a hypergraph* H' *from* H *by unifying the vertices* $v_1, \dots, v_t$. *In particular* $V(H') = V(H) \setminus \{v_i \mid 2 \leq i \leq t\}$ *and for every hyperedge* $e \in E(H)$ *make the edge* $e' \in E(H')$ *where* $e' = e$ *if* $v_i \notin e$ *for every* $i \in [t]$, *and* $e' = (V(e) \setminus \{v_i \mid 2 \leq i \leq t\}) \cup \{v_1\}$ *otherwise. We obtain* (H', k).

Our next reduction rule takes care of the case when the hypergraph has a strong cut-set. While this reduction rule first appeared in [149], for the sake of completeness, we wish to present here its complete proof. The rest of this section is devoted to the proof of the following.

Theorem 9.5 ([149])**.** *There is a polynomial time algorithm that given a strong cut-set* X *of a connected hypergraph* $H = (V(H), E(H))$ *finds a cut-set* $X' \subseteq X$ *such that* $X' \neq \emptyset$ *and there exists a coloring* χ *such that* $|\mathbf{split}(\chi, H)| = \mathbf{split}(H)$ *and* χ *splits all the hyperedges in* X'. *In fact, it shows that given any coloring* c, *there exists a coloring* χ *such that* $\mathbf{split}(\chi, H) = \mathbf{split}(c, H) \cup X'$.

We first five an algorithmic version of Proposition 9.1 which is central to the proof of Theorem 9.5. Given a forest $\mathcal{F} = (F, g)$ we classify the edges of $E(H)$ as follows. An edge $e \in E(H)$ is

- a *forest edge* if there exists an edge f in $E(F)$ such that $g(f) = e$,
- a *cut edge* if there exist two connected components C_1 and C_2 of F such that $V(e) \cap V(C_1) \neq \emptyset$ and $V(e) \cap V(C_2) \neq \emptyset$,
- an *unused edge* if there does not exist an edge f in $E(F)$ such that $g(f) = e$; that is e is not in the image of the map g.

We remark that an edge e can be a forest edge as well as a cut edge at the same time. Similarly an edge can be a cut edge as well as an unused edge at the same time.

Definition 9.3. For a hypergraph $H = (V(H), E(H))$, a forest $\mathcal{F} = (F, g)$ and $e_1, e_2 \in E(H)$, we say that an edge e_2 follows e_1 if e_1 is a forest edge of F and e_2 is a cut edge with respect to $\mathcal{F}' = (F', g')$ where $F' = (V(H), E(F) \setminus \{g^{-1}(e_1)\})$ and $g'(f) = g(f)$ for $f \in E(F')$.

We are now in position to state and prove the algorithm version of Proposition 9.1 which will be used later on.

Theorem 9.6 ([149])**.** *There is a polynomial time algorithm that given a connected hypergraph $H = (V(H), E(H))$ and a forest $\mathcal{F} = (F, g)$ of H such that $|E(F)| < |V(H)| - 1$ finds either a forest $\mathcal{F}' = (F', g')$ of H with $|E(F')| \geq |E(F)| + 1$ or a strong cut-set X of H.*

Proof. Given a hypergraph $H = (V(H), E(H))$ and a forest $\mathcal{F} = (F, g)$ of H, a sequence of hyperedges $\mathcal{L}(H, \mathcal{F}) = e_1 e_2 \dots e_t$ such that $e_i \in E(H)$ is called an *augmenting sequence* if (a) e_1 is a cut edge, (b) e_{i+1} follows e_i for all $1 \leq i \leq t$, and (c) e_t is an unused edge.

We first prove that if there exists an augmenting sequence with respect to a forest $\mathcal{F} = (F, g)$ of H then there exists a forest $\mathcal{F}' = (F', g')$ of H with $|E(F')| \geq |E(F)| + 1$. We prove this by induction on the length of the shortest augmenting sequence t. If $t = 1$ then $\mathcal{L}(H, \mathcal{F}) = e_1$. In this case e_1 is a cut edge as well as an unused edge. Since e_1 is a cut edge there exists two connected components C_1 and C_2 of F such that $V(C_1) \cap V(e_1) \neq \emptyset$ and $V(C_2) \cap V(e_1) \neq \emptyset$. Let $u \in V(C_1) \cap V(e_1)$ and $v \in V(C_2) \cap V(e_1)$. Now define $F' = (V(H), E(F) \cup \{\{u, v\}\})$ and $g' : E(F') \to E(H)$ as $g'(f) = g(f)$ if $f \in E(F)$ and $g'(\{u, v\}) = e_1$. Since we have added an edge between two distinct components of F, we have that the F' is also a forest and has one more edge than that in F.

Assume now that $t \geq 2$ and that if we are given a forest $\mathcal{F} = (F, g)$ of H and a shortest augmenting sequence $\mathcal{L}(H, \mathcal{F})$ of length at most $t' < t$ then there exists a forest $\mathcal{F}' = (F', g')$ of H with $|E(F')| \geq |E(F)| + 1$. Let $\mathcal{F} = (F, g)$ be a forest and $\mathcal{L}(H, \mathcal{F}) = e_1 e_2 \ldots e_t$ be a shortest augmenting sequence of F with length t. Observe that e_1 is a cut edge. Hence there exist two connected components C_1 and C_2 of F such that $V(C_1) \cap V(e_1) \neq \emptyset$ and $V(C_2) \cap V(e_1) \neq \emptyset$. Let $u \in V(C_1) \cap V(e_1)$ and $v \in V(C_2) \cap V(e_1)$. Furthermore e_1 is a forest edge and hence we let $f \in E(F)$ such that $g(f) = e_1$. Now we construct a forest $\mathcal{F}^* = (F^*, g')$ as follows. We let $F^* = (V(H), E(F) \cup \{\{u, v\}\} \setminus \{f\})$ and $g' : E(F^*) \to E(H)$ as $g'(f) = g(f)$ if $f \in E(F)$ and $g'(\{u, v\}) = e_1$. Now we show that the $\mathcal{L}(H, \mathcal{F}^*) = e_2 \ldots e_t$ is an augmenting sequence of length at most $t - 1$ for $\mathcal{F}^*$.

To show this we will use the following properties of $\mathcal{L}(H, \mathcal{F})$.

- if $t > 1$ then e_1, e_2, ..., e_{t-1} are forest edges;
- if $t > 1$ then e_2, ..., e_{t-1} are not cut edges with respect to $\mathcal{F}$.

The first property follows from the definition of $\mathcal{L}(H, \mathcal{F})$ and the second from the choice of $\mathcal{L}(H, \mathcal{F})$ to be a shortest augmenting sequence.

We first show that e_2 is a cut edge with respect to $\mathcal{F}^*$. Indeed, recall that e_2 is not a cut edge with respect to $\mathcal{F}$. Then there exists exactly one connected component C of $\mathcal{F}$ such that $V(e_2) \cap C \neq \emptyset$ and therefore $V(e_2)$ intersects with at most one of the C_1 and C_2. However, as e_2 follows e_1, then there are connected components C' and C'' of $E(F) \setminus \{f\}$ such that $V(e_2) \cap C' \neq \emptyset$ and $V(e_2) \cap C'' \neq \emptyset$. This implies that $C = C' \cup C'' \cup \{f\}$. It is easy to see that if $C \neq C_1$ and $C \neq C_2$ then C' and C'' are also connected components of $\mathcal{F}^*$ and therefore e_2 is a cut edge of $\mathcal{F}^*$. In the case where $C = C_1$ ($C = C_2$, respectively) then without loss of generality assume that $u \in V(C')$ ($v \in V(C')$, respectively). This implies that $C' \cup C_2 \cup \{\{u, v\}\}$ ($C' \cup C_1 \cup \{\{u, v\}\}$, respectively) and C'' are connected components of $\mathcal{F}^*$ and therefore e_2 is a cut edge of $\mathcal{F}^*$.

Notice now that $g(E(F)) = g'(E(F^*))$, thus we have that e_t is an unused edge of $\mathcal{F}^*$. The only thing that remains to be proved is that e_{j+1} follows e_j for $j \in \{2, \ldots, t-1\}$ with respect to $\mathcal{F}^*$. Let e_j be a hyperedge, $j \in \{2, \ldots, t-1\}$, and let u_1 and u_2 be two vertices in $V(e_{j+1})$ that lie in different connected components of $F \setminus g^{-1}(e_j)$. We prove that u_1 and u_2 lie in different connected components of $F^* \setminus g'^{-1}(e_j)$. Suppose not, then there is a path P from u_1 to u_2 in $F^* \setminus g'^{-1}(e_j)$. If P does not contain $g'^{-1}(e_1)$ then P is a path from u to v in $F \setminus g^{-1}(e_j)$ because $g'^{-1}(e_j) = g^{-1}(e_j)$ and $g'^{-1}(e_1)$ is the only edge in F^* that is not in F. This contradicts that e_{j+1} follows e_j with respect to $\mathcal{F}$. If P contains $g'^{-1}(e_1)$ then u_1 and u_2 must lie in *different* connected components of $F^* \setminus g'^{-1}(e_1) = F \setminus g^{-1}(e_1)$. But, as $\mathcal{L}(\mathcal{H}, \mathcal{F})$ is of shortest length, e_{j+1} does not follow e_1 with respect to $\mathcal{F}$, and hence u_1 and u_2 must lie in the *same* connected component $F \setminus g^{-1}(e_1)$, a contradiction. Thus, we conclude that e_{j+1} follows e_j with respect to $\mathcal{F}^*$.

Hence we have shown that $\mathcal{L}(H, \mathcal{F}^*) = e_2 \ldots e_t$ is an augmenting sequence of length at most $t-1$ for $\mathcal{F}^*$. This implies that $\mathcal{F}^*$ has a shortest augmenting sequence of length at most $t-1$ and hence by the induction hypothesis this implies that there exists a forest $\mathcal{F}' = (F', g')$ of H with $|E(F')| \geq |E(F^*)| + 1 = |E(F)| + 1$.

For the other part of the proof we show that if we do not have an augmenting sequence then we have a strong cut-set. We say that a hyperedge e is *reachable* from a hyperedge $e^* \in Y$ if there exists a sequence of hyperedges $e^* e_1 \ldots e_l e$ and e_1 follows e^*, e_{i+1} follows e_i for $i \in \{1, \ldots, l-1\}$, and e follows e_l. Let Y be the set of cut edges with respect to $\mathcal{F}$ and X be the set of all hyperedges containing Y and all those hyperedges which are reachable from a hyperedge in X. We claim that X is the desired strong cut-set. Let $H' = (V(H), E(H) \setminus X)$ be the hypergraph obtained from H by removing the hyperedges from X. Now we show $P(H')$ has at least $|X| + 2$ connected components. Observe that all the edges in X are forest edges, otherwise there would exist an augmenting sequence. Let $X^{-1} = \{g^{-1}(x) \mid x \in X\}$. The forest F which we started with has at least two components and hence when we remove the edges from X^{-1} it has at least $|X| + 2$ connected components. We show that for every connected component C, of $F' = (V(H), E(F) \setminus X^{-1})$, $P(H')[V(C)]$ is a connected component. Suppose not, then there exist two connected components C_1 and C_2 of F' such that there exists a hyperedge $e \notin X$ such that $V(e) \cap V(C_1) \neq \emptyset$ and $V(e) \cap V(C_2) \neq \emptyset$. Let $u \in V(e) \cap V(C_1)$ and $v \in V(e) \cap V(C_2)$. Since e is not a cut edge, u and v are in the same component of F. Since u and v are not in the same component of F' there is a hyperedge $e' \in X$ such that u and v are in different components of $F \setminus g^{-1}(e')$. Hence e follows e', a contradiction.

We have proved that for a connected hypergraph $H = (V(H), E(H))$ and a forest $\mathcal{F} = (F, g)$ of H such that $|E(F)| < |V(H)| - 1$, either there exists a forest $\mathcal{F}' = (F', g')$ of H with $|E(F')| \geq |E(F)| + 1$ or there exists a strong cut-set X of H. We can make our proof constructive if we have a way to find a shortest augmenting path with respect to $\mathcal{F}$. In what follows we show how to find a shortest augmenting path corresponding to $\mathcal{F}$ by reducing this to finding a shortest path in an auxiliary directed graph. We define a graph G' with vertex set $V(G') = \{v_e \mid e \in E(H)\}$, that is, for every hyperedge $e \in E(H)$ we add a vertex v_e to $E(G)$. We add an edge from v_e to v_f if f follows e with respect to $\mathcal{F}$. Hence to find a shortest augmenting sequence it is sufficient to do a breadth first search in G' starting from $\{v_e \in V(G') \mid e \text{ is a cut edge}\}$ and checking whether a vertex v_f corresponding to an unused hyperedge is reached. It is clear that this procedure takes polynomial time. If we find an augmenting sequence then we can find the desired forest $\mathcal{F}'$ in polynomial time and if we do not find an augmenting sequence then we can find the desired cut-set X as described in the proof in polynomial time. $\square$

Given a graph G a *matching* M in G is a set of edges where no two of them share a vertex and a set $S \in V(G)$ is *saturated* by M if for every $v \in S$ there exists an edge $e \in M$ such that $v \in e$.

We may now proceed to the proof of Theorem 9.5.

Proof of Theorem 9.5. Let $H^* = (V(H), E(H) \setminus X)$ and let $|X| = t$. By assumption, X is a strong cut-set and hence the primal graph $P(H^*)$ has at least $t+2$ connected components. Let the connected components of $P(H^*)$ be $\mathcal{C} = \{C_1, \ldots, C_q\}$ where $q \geq t+2$ and $X = \{e_1, \ldots, e_t\}$. We construct an auxiliary bipartite graph $\mathcal{B}$ with vertex set $A \cup B$ with a vertex $a_i \in A$ for every edge $e_i \in X$ and a vertex $b_i \in B$ for every connected component $C_i \in \mathcal{C}$. There is an edge $\{a_i, b_j\}$ if $V(e_i) \cap V(C_j) \neq \emptyset$.

We prove the statement of the lemma by induction on $|X|$. For the base case we assume that $|X| = 1$ and $X = \{e_1\}$. In particular, we show that given any $f : V(H) \to \{-1, 1\}$ there exists a function $g : V(H) \to \{-1, 1\}$ such that $\mathbf{split}(g) = \mathbf{split}(f) \cup \{e_1\}$ which will prove the desired assertion. If $e_1 \in \mathbf{split}(f)$ the statement follows, so assume that $e_1 \notin \mathbf{split}(f)$. Since $P(H)$ is connected we have that $\{a_1, b_j\}$, $j \in \{1, \ldots, q\}$ are edges in $\mathcal{B}$. Let $g : V(H) \to \{-1, 1\}$ be such that $g(v) = f(v)$ if and only if $v \notin C_1$. That is, for all vertices in C_1, g flips the assignment given by f.

Observe that $e_1 \in \mathbf{split}(g)$ since $V(e_1)$ contains a vertex $u \in C_1$ and a vertex $v \in C_2$. Since $f(u) = f(v)$, $g(u) \neq g(v)$ and hence $e_1 \in \mathbf{split}(g)$. As X is a strong cut-set of H and $X = \{e_1\}$ for every edge e in $\mathbf{split}(f)$ we have that $V(e)$ is completely contained in one of the components and hence, $e \in \mathbf{split}(f)$ implies $e \in \mathbf{split}(g)$. This completes the proof for the base case. So we assume that $|X| \geq 2$ and that the statement of the lemma holds for all X' satisfying the conditions of the lemma and $|X'| < |X|$. In the inductive step we consider two cases:

(a) there does not exist a matching in $\mathcal{B}$ which saturates A; or

(b) there is a matching saturating A in $\mathcal{B}$.

In Case (a), by Hall's theorem [114], we know that there exists a subset $A' \subseteq A$, $A' \neq \emptyset$ such that $|A'| > |N(A')|$ and such a set can be found in polynomial time. We claim that $X' = X \setminus \{e_j \mid a_j \in A'\}$ is a strong cut-set and is of smaller size than X. It is clear that $|X'| < |X|$ as $A' \neq \emptyset$. We now show that X' is indeed a strong cut-set. Let $\mathcal{C}' = \mathcal{C} \setminus \{C_j \mid b_j \in N(A')\}$. Observe that in $H' = (V(H), E(H) \setminus X')$, every $C_i \in \mathcal{C}'$ is a connected component. The size of $\mathcal{C}'$ is bounded as follows

$$|\mathcal{C}'| = |\mathcal{C}| - |N(A')| \geq (t+2) - |N(A')| > (t+2) - |A'| = t - |A'| + 2 = |X'| + 2,$$

and hence X' is indeed a strong cut-set. In this case the statement of the lemma follows from the induction hypothesis as $|X'| < |X|$.

For Case (b) we assume that we have a matching M saturating A. Without loss of generality, let M be $\{\{a_1, b_1\}, \ldots, \{a_t, b_t\}\}$. Let $U = \{b_{t+1}, \ldots, b_q\}$ be the set of vertices in B that are not saturated by M. Iteratively we construct a set U' containing U as follows. Initially we set $U' := U$ and $\tilde{A} = A$.

- Check whether there exists a neighbor of a vertex in U' in $\tilde{A}$; if yes go to the next step. Otherwise, output U'.
- Let a_j be a vertex in $\tilde{A}$ having a neighbor in U'. Set $U' := U' \cup \{b_j\}$ (b_j is the matching endpoint of a_j in B), $\tilde{A} := \tilde{A} \setminus \{a_j\}$ and go to the first step.

Let U' be the set returned by the iterative process above. Observe that, as $P(H)$ is connected, there exists at least one vertex a_j having a neighbor in U. Therefore the above process enters the second step at least once and hence $U \subsetneq U'$ and $\tilde{A} \subsetneq A$. Let $A' = A \setminus \tilde{A}$ and let $X' = \{e_j \mid a_j \in A'\}$. In what follows we prove that X' is the desired subset of X mentioned in the statement of the lemma.

We first show that X' is a strong cut-set. Let $\mathcal{C}' = \{C_j \mid b_j \in U'\}$. Notice that, from the construction, there is no vertex in $\tilde{A}$ that has a neighbor both in U' and $B \setminus U'$. This implies that every $C_i \in \mathcal{C}'$ is a connected component of $H' = (V(H), E(H) \setminus X')$. The size of C' is bounded as follows

$$|\mathcal{C}'| = |U'| = |U| + |A'| \geq |A'| + 2 = |X'| + 2,$$

and hence X' is a strong cut set.

We show that given any $f : V(H) \to \{-1, 1\}$ there exists a function $g : V(H) \to \{-1, 1\}$ such that $\mathbf{split}(g) = \mathbf{split}(f) \cup X'$. This will complete the proof of the theorem. Let $U' \setminus U = \{b_{1'}, b_{2'}, \ldots, b_{r'}\}$ and without loss of generality assume that $b_{1'}$, $b_{2'}$, ..., $b_{r'}$ is the order in which these elements are included in the set U'. Let $\mathcal{B}_i = \mathcal{B}[U \cup \{b_{1'}, \ldots, b_{i'}\} \cup \{a_{1'}, \ldots, a_{i'}\}]$ and H_i be the hypergraph induced by the vertices in $V(C_{1'}) \cup \cdots \cup V(C_{i'}) \cup e_{1'} \cup \cdots \cup e_{i'}$. Iteratively we construct the function $g : V(H) \to \{-1, 1\}$ as follows. Initially we set $g := f$ and $i := 1$ and repeat the following until $i = r$:

> Check whether $e_{i'} \in \mathbf{split}(g, H_i)$. If yes $i := i + 1$ and repeat. Otherwise let $C_{i'}$ be the connected component corresponding to $b_{i'}$ having vertex set $V(C_{i'})$. Now for every vertex $u \in V(C_{i'})$ change $g(u)$ to $-f(u)$. Basically, we flip the assignment of -1 and 1 in the vertex set $V(C_{i'})$. Set $i := i + 1$ and repeat the procedure.

Now we show that $\mathbf{split}(g) = \mathbf{split}(f) \cup X'$. Notice that if $e \notin X$ then, as X is a strong cut-set of H, there exists a connected component $C \in \mathcal{C}$ such that $e \subseteq C$. Notice then that either $g(v) = f(v)$ for every $v \in e$ or $g(v) = -f(v)$ for every $v \in V$. By combining this with Observation 9.1 we obtain that $e \in \mathbf{split}(f)$ if and only if $e \in \mathbf{split}(g)$. Notice now that, by the way X' was constructed, if $e \in X \setminus X'$ then $V(e) \cap V(C_{i'}) = \emptyset$, $i \in [r]$. Observe that $g(v) = f(v)$, $v \in V(H) \setminus (V(C_{1'}) \cup \cdots \cup V(C_{r'}))$ and hence $e \in \mathbf{split}(g)$ if and only if $e \in \mathbf{split}(f)$. It remains to show that if $e \in X'$, then $e \in \mathbf{split}(g)$. Notice that if, at step i, $e_i' \notin \mathbf{split}(g, H_i)$ then every vertex is assigned the same color by g. Furthermore, notice that by construction there exists a $b_{j'} \in \{b_{1'}, b_{2'} \ldots, b_{(i-1)'}\}$ such that $V(e_{i'}) \cap V(C_{j'}) \neq \emptyset$ and recall that $e_{i'} \cap V(C_{i'}) \neq \emptyset$. As the color of the vertices in $V(C_{i'})$ changes we obtain that after this step of the algorithm completes $e_{i'}$ is split. Finally, observe that for every $q < i$, $e_{q'} \in \mathbf{split}(g, H_{q'})$ and $H_{q'} \subseteq H_{i'} \setminus V(C_{i'})$. This implies that all the edges $e_{q'}$, with $q < i$, remain split after flipping the colors of the vertices in $V(C_{i'})$. Thus after the r^{th} step of the procedure we have that $\mathbf{split}(g) = \mathbf{split}(f) \cup X'$. This concludes the proof. $\square$

Notice now that Theorem 9.5 results in the following reduction rule.

Reduction Rule 2. : *Let (H, k) be an instance of* AA-r-SS *and X' be a set as defined in Theorem 9.5. Remove X' from the set of hyperedges and reduce k to $k - \sum_{e\in X'} \frac{1}{2^{|e|-1}}$, that is, obtain an instance $(H', k - \sum_{e\in X'} \frac{1}{2^{|e|-1}})$, where $E(H') = E(H) \setminus X'$.*

Now we argue the correctness of Reduction Rule 2. Let (H, k) be an instance of AA-r-SS and X' be as in the Theorem 9.5. By Theorem 9.5 we know that there exists a coloring χ such that

$$|\mathbf{split}(\chi, H)| = \mathbf{split}(H)$$

and χ splits all the hyperedges in X'. This implies that in H' at least

$$\mu_H + k - |X'| \geq \mu_{H'} + \sum_{e\in X'} \left(1 - \frac{1}{2^{|e|-1}}\right) + k - |X'| \geq \mu_{H'} + k - \sum_{e\in X'} \frac{1}{2^{|e|-1}}$$

hyperedges are split. For the other direction observe that if in H' we have $\mu_{H'} + k - \displaystyle\sum_{e\in X'} \frac{1}{2^{|e|-1}}$ hyperedges split then in H we have $\mu_{H'} + k - \displaystyle\sum_{e\in X'} \frac{1}{2^{|e|-1}} + |X'|$ hyperedges split. The last inequality implies that in H we have $\mu_H + k$ split hyperedges. This proves the correctness of the Reduction Rule 2.

9.4 Linear Kernel for Fixed r and Proof of Theorem 9.2

In this section, we combine our results from the previous section with known reduction rules obtained in [149] for p-Set Splitting (that we proved in the previous section) to obtain the desired kernel for AA-r-SS when $r = O(1)$. Finally, we give the proof of Theorem 9.2.

Theorem 9.7. *For a fixed r,* AA-r-SS *admits a kernel with $O(k)$ vertices.*

Proof. Let (H, k) be a reduced instance of AA-r-SS, that is, we cannot apply Reduction Rules 1 and 2. As Reduction Rule 1 does not apply, H is connected. Moreover, as Reduction Rule 2 does not apply H does not have a strong cut-set. From Theorem 9.1, it follows that if $k \leq \frac{n-1}{r \cdot 2^{r-1}}$ then it is a YES-instance. Otherwise, $\frac{n-1}{r \cdot 2^{r-1}} \leq k$, thus $n \leq r \cdot 2^{r-1}k + 1 = O(k)$. □

Proof of Theorem 9.2. As in the proof of Theorem 9.7 we assume that (H, k) is a reduced instance and hence H is partition connected. For the simplicity of an argument choose $\alpha = 1/2$ and thus $r = \log \sqrt{n}$. From Theorem 9.1, it follows that if $k \leq \frac{n-1}{r \cdot 2^{r-1}}$ then it is a YES-instance. Otherwise, $\frac{n-1}{r \cdot 2^{r-1}} \leq k$, thus $n \leq r \cdot 2^{r-1}k + 1$. Substituting $r = \log \sqrt{n}$, we get that $2n \leq (\log \sqrt{n})\sqrt{n}k + 1$. This implies that $k \geq n^{\frac{1}{2} - \epsilon}$ for every fixed $\epsilon > 0$. Since we can always solve AA-r-SS for any r in time 2^n, we get that AA-$\alpha \log n$-SS can be solved in time $O^*(2^{k^{\frac{2}{1-\epsilon}}})$. We remark that we could have chosen $\alpha = 1 - \delta$ for any fixed constant δ. □

9.5 Lower Bound Result and Proof of Theorem 9.3

In this section, we will show that AA-$\lceil \log n \rceil$-SS is not in XP unless $\mathsf{NP} \subseteq \mathsf{DTIME}\ [n^{\log \log n}]$. Towards this we will give a suitable reduction from r-NAE-SAT. A r-CNF formula $\phi = C_1 \wedge \cdots \wedge C_m$ is a boolean formula where each clause has size at least 2 and at most r and each clause is a disjunction of literals. r-NAE-SAT is a variation of r-SAT, where given a r-CNF formula $\phi = c_1 \wedge \cdots \wedge c_m$ on n variables, say $V(\phi) = \{x_1, \ldots, x_n\}$, the objective is to find a $\{0,1\}$-assignment to $V(\phi)$ such that all the clauses get split. An assignment splits a clause if at least one of its literals gets the value 1 and at least one of its literals gets the value 0. We call an assignment that splits every clause a *splitting assignment.*

Proof of Theorem 9.3. We will prove the theorem in three steps. First, we prove that r-NAE-SAT is NP-complete for $r = \lceil \log n \rceil + 1$. Throughout this proof r is set to $\lceil \log n \rceil + 1$. It is known that $\lceil \log n \rceil$-SAT is NP-complete even when the input has at most cn clauses [40].

In order to show that r-NAE-SAT is NP-complete we will give the standard reduction to it from $\lceil \log n \rceil$-SAT. Let $\phi = C_1 \wedge \cdots \wedge C_m$ be an instance to $\lceil \log n \rceil$-SAT. We introduce a new global variable z and create an instance $\phi' = C'_1 \wedge \cdots \wedge C'_m$ for r-NAE-SAT by taking $C_i \wedge z$ for each clause C_i of ϕ. If there is a satisfying assignment π of ϕ then at least one literal in each clause of ϕ is set to 1. Now, $\pi \cup \{z = 0\}$ is an assignment satisfying ϕ' and splitting each clause. Conversely, if τ is a splitting satisfying assignment for ϕ' then so is the complementary assignment τ' (the one obtained from τ by flipping 0's to 1 and 1's to 0's). In one of the two assignments, say τ, $z = 0$ and some other literal in each clause is set to 1. Thus the restriction of τ to the variables of ϕ is a satisfying assignment for ϕ.

Our second step is to show a many one reduction from r-NAE-SAT to r-SET-SPLITTING running in time $O(n^{\log \log n})$. In r-SET-SPLITTING we are given an r-hypergraph and the objective is to check whether there exists a splitting hypergraph 2-coloring (a hypergraph 2-coloring that splits every hyperedge). Recall, we defined hypergraph 2-coloring as a function from $V(H)$ to $\{-1, 1\}$. However, for this proof we will make it a function from vertex set of H to $\{0, 1\}$.

Let $\phi = C_1 \wedge \cdots \wedge C_m$ be an instance of r-NAE-SAT, with n sized variable set $V(\phi)$ and $m \leq \gamma n$ clauses, for some fixed constant γ. We want to construct an instance H of r-Set-Splitting. For each variable x of ϕ, H has two vertices x_1 and x_2. For each clause C of ϕ we construct a corresponding hyperedge C_h of H in the following way: If a variable x appears in the positive form in C then we include x_1 in C_h and if $\neg x$ appears in C then we include x_2 in C_h. Essentially, x_1 corresponds to x and x_2 to $\neg x$.

Now, we describe an algorithm that, given a variable x of ϕ, outputs a collection $\mathcal{S}^x$ of hyperedges of size r. Let $\mathcal{S}_1^x = \{\{x_1, x_2\}\}$. For $i = 2, \ldots, r-1$, we do as follows.

> We let $\mathcal{S}_i^x = \mathcal{S}_{i-1}^x$ and, while there exists a hyperedge $S \in \mathcal{S}_i^x$ such that $|S| = i$, introduce r new vertices w_j^S, $j \in [r]$, add the sets $\{w_1^S, w_2^S, \ldots, w_r^S\}$ and $S \cup \{w_j^S\}$, $j \in [r]$, in $\mathcal{S}_i^x$ and remove the set S.

Add the hyperedges of $\mathcal{S}^x = \mathcal{S}_{r-1}^x$ to the hypergraph H. Notice that after the execution of step $i \in [r-1]$ all the hyperedges contained in the set $\mathcal{S}_i^x$ are of size either $i+1$ or r. Thus, every hyperedge in $\mathcal{S}_{r-1}^x$ has size exactly r.

Summarizing the construction we have following set of hyperedges:

$$\{C_h \mid C \text{ a clause of } \phi\} \cup \Big(\bigcup_{x \in V(\phi)} \mathcal{S}^x \Big).$$

The vertex set of H consists of vertices appearing in these clauses.

It is easy to see that for any splitting assignment τ of ϕ we have a splitting $\{0,1\}$-coloring of H by setting $x_1 = \tau(x)$ and $x_2 = \tau(\neg x)$ and by assigning 0 or 1 to newly added vertices in a way that splits all the hyperedges that do not contain vertices that correspond to variables of ϕ. We can assign 0 or 1 to newly added vertices in consistent way because all the hyperedges that do not contain vertices that correspond to variables of ϕ are pairwise disjoint.

To prove the converse we show the following auxiliary claim. The claim will allow us to argue that in any splitting 2-coloring of H the vertices x_1 and x_2 corresponding to a variable x will always be assigned either 0 and 1, respectively, or 1 and 0, respectively.

Claim 13. *For every $i \in \{2, \ldots, r-1\}$ and every 2-coloring χ that splits all the hyperedges in $\mathcal{S}_i^x$, there are hyperedges $S_1, S_2 \in \mathcal{S}_i^x$ such that $\{x_1, x_2\} \subseteq S_1 \cap S_2$, $|S_1| = |S_2| = i+1$, and $S_i' = S_i \setminus \{x_1, x_2\}$, $i \in [2]$, are monochromatic and $\chi(S_1') \cap \chi(S_2') = \emptyset$ (that is, the colors on vertices of S_1' and S_2' are disjoint).*

Proof. We prove this claim using induction on i. For base case let $i = 2$. In this case we know that $\mathcal{S}_2^x$ is precisely $\{w_1^S, w_2^S, \ldots, w_r^S\}$ and $S \cup \{w_j^S\}$, $j \in [r]$, where $S = \{x_1, x_2\}$. Since $\{w_1^S, w_2^S, \ldots, w_r^S\}$ is split we know that there exist $a, b \in [r]$, such that $\chi(w_a^S) = 0$ and $\chi(w_b^S) = 1$. Take $S_1 = S \cup \{w_a^S\}$ and $S_2 = S \cup \{w_b^S\}$. Clearly, this satisfies all the properties described in the statement of the claim.

Assume now that the claim holds for $\ell = i \geq 2$. Let χ be a splitting 2-coloring of $\mathcal{S}_{\ell+1}^x$. To apply induction we would like to show that χ', the restriction of χ to the vertices of $\mathcal{S}_\ell^x$, is a splitting 2-coloring of $\mathcal{S}_\ell^x$. However, new vertices are added in $\mathcal{S}_{\ell+1}^x$ only when there exists a hyperedge $S \in \mathcal{S}_\ell^x$ such that $|S| = \ell$. And for this case we introduced r new vertices w_j^S, $j \in [r] = \{1, \ldots, r\}$, added the sets $\{w_1^S, w_2^S, \ldots, w_r^S\}$ and $S \cup \{w_j^S\}$, $j \in [r]$, in $\mathcal{S}_i^x$ and removed the set S. The set $\{w_1^S, w_2^S, \ldots, w_r^S\}$ in $\mathcal{S}_{\ell+1}^x$ forces that the vertices in S can not be assigned all 0 or all $1's$ under χ and thus, under χ'. This proves that χ' splits all the sets of size ℓ in $\mathcal{S}_\ell^x$. Furthermore all other sets in $\mathcal{S}_i^x$ are of size r and hence are also present in

$\mathcal{S}^x_{\ell+1}$ and thus automatically get split by χ'. This proves that χ' is indeed a splitting 2-coloring for $\mathcal{S}^x_\ell$.

Thus, from the inductive hypothesis there are hyperedges Q_1 and Q_2 of size ℓ such that $Q_j \setminus \{x_1, x_2\}$, $j \in [2]$, are monochromatic but not with the same color. Notice now that there are vertices $w_p^{Q_1}$, $p \in [r]$, and $w_q^{Q_2}$, $q \in [r]$, such that the hyperedges $\{w_1^{Q_1}, \dots, w_r^{Q_1}\}$, $\{w_1^{Q_2}, \dots, w_r^{Q_2}\}$, $Q_1^p = Q_1 \cup \{w_p^{Q_1}\}$, $p \in [r]$, and $Q_2^q = Q_2 \cup \{w_q^{Q_2}\}$, $q \in [r]$, all appear in $\mathcal{S}^x_{\ell+1}$ and are split by χ. Therefore, there are vertices $w_{p1}^{Q_1}, w_{p2}^{Q_1}$, and $w_{q1}^{Q_2}, w_{q2}^{Q_2}$ such that $\chi(w_{p1}^{Q_1}) \neq \chi(w_{p2}^{Q_1})$ and $\chi(w_{q1}^{Q_2}) \neq \chi(w_{q2}^{Q_2})$. This implies that we can select $S_1 = Q_1^{p_j}$ and $S_2 = Q_2^{q_s}$, $j, s \in [2]$, such that $Q_1^{p_j} \setminus \{x_1, x_2\}$ and $Q_2^{q_s} \setminus \{x_1, x_2\}$ are monochromatic but not with the same color. This completes the proof of the claim. □

Now we give the proof of the reverse direction. Let χ be a splitting $\{0,1\}$-coloring of H. Then by Claim 13 we get that each pair (x_1, x_2) is colored differently. The last assertion is easy to see as the only way we can split S_1 and S_2 coming from Claim 13 is by assigning x_1 and x_2 different colors for every variable $x \in V(\phi)$ under χ. By setting $x = \chi(x_1)$ for each variable of ϕ we get a splitting assignment for ϕ.

From the construction, it follows that the total number of vertices in H is at $2n + nT$, where $T = \frac{r(r^{r-2}-1)}{r-1}$, and the total number of hyperedges in H is $\gamma n + n(T+1)$. Therefore, the number of hyperedges is linear in the number of vertices of the hypergraph. Since $r = \lceil \log n \rceil + 1$ we have that the time to construct our instance will take $(\log n)^{O(\log n)} = n^{O(\log \log n)}$.

For the third step we will show that if we have an algorithm for AA-($\lceil \log n \rceil + 1$)-SS running in time $n^{g(k)}$ then we can solve ($\lceil \log n \rceil + 1$)-Nae-Sat with n variables and γn clauses (the number of clauses is linear in number of variables) in $n^{O(\log \log n)}$ time. This is contradictory unless $\mathsf{NP} \subseteq \mathrm{DTIME}(n^{\log \log n})$.

Given an instance ϕ of $(\lceil \log n \rceil + 1)$-NAE-SAT with n variables and γn clauses we first apply the reduction to $(\lceil \log n \rceil + 1)$-SET-SPLITTING. This gives us an instance H with n' vertices and $\delta n'$ hyperedges for some fixed constant δ. However, $\mu_H = (1 - \frac{1}{2^{\log n'}}) \cdot \delta n' = \delta n' - \delta$. Thus, we can check whether H has a splitting 2-coloring by taking $k = \lceil \delta \rceil$ as a parameter in AA-$(\lceil \log n \rceil + 1)$-SS and running the algorithm for AA-$(\lceil \log n \rceil + 1)$-SS. Since $k = \lceil \delta \rceil$ is a constant, the algorithm with running time $n^{g(k)}$ runs in polynomial time. This completes the proof.

Thus, we have shown that there exists a fixed constant β such that AA-$\beta \log n$-SS is not XP unless NP$\subseteq$ DTIME$(n^{\log \log n})$. This completes the proof of the theorem. $\square$

9.6 Conclusions

In this chapter, we generalized an old result by Edwards on the size of a max cut on connected graphs to partition connected r-hypergraphs. We then used this result to show an above guarantee version of MAX r-SET SPLITTING FPT. There are several interesting problems that are still open in parameterized study of problems above guaranteed lower bounds, as well as in the specific directions pursued in this paper. Most notable ones are:

- Is the lower bound of $\mu_H + \frac{n-1}{r2^{r-1}}$ on $\zeta(H)$ for partition connected r-hypergraphs tight? That is, is there an infinite family of partition connected r-hypergraphs where $\zeta(H) = \mu_H + \frac{n-1}{r2^{r-1}}$?

- Is $\lceil \log n \rceil$-SET-SPLITTING with linear number of clauses NP-complete?

- Is the question of finding an independent set of size $\frac{n}{4} + k$ on planar graphs FPT? Even obtaining an algorithm in XP remains elusive.

CHAPTER 10

Graph Searching: A Game Characterization of Cycle-rank

10.1 Introduction

Graph searching games are increasingly becoming a popular way to characterize, and even define, practical graph parameters. There are many advantages to a characterization by graph searching games: it provides a useful intuition which can assist in constructing more general or more specific parameters; it gives insights into relations with other, similarly characterized parameters; and it is particularly useful from an algorithmic perspective as many parameters associated with such games are both structurally robust and efficiently computable.

10.1.1 Node Search in Graphs

One of the most common graph searching games is the node-search game. In this game several *searchers* and one *fugitive* occupy vertices of the graph and make simultaneous moves. The (*omniscient*) fugitive moves along searcher-free paths of arbitrary length whereas the searchers' movements are not constrained by the topology of the graph. The goal of the game is to minimize the number of searchers required to capture the fugitive by cornering him in some part of the graph and placing a searcher on the same vertex.

This game has been extensively studied and several important graph parameters such as tree-width [56, 217] and path-width [137] can be characterized by natural variants of this game. One variation frequently used, indeed the one which separates tree-width and path-width, is whether the location of the fugitive is known or unknown to the searchers.

Another common variation is whether the searchers use a monotone or a non-monotone searching strategy, that is, whether their strategy provides to the fugitive access to already searched areas (non-monotone strategy) or not (monotone strategy). Monotone search strategies lead to algorithmically useful decompositions, whereas non-monotone strategies are more robust under graph operations and hence reflect structural properties. Therefore, showing that monotone strategies require no more searchers than non-monotone strategies is an important and common question in the area. Whilst node-search games on undirected graphs tend to enjoy monotonicity [20, 145, 217], on digraphs the situation is much less clear [2, 16, 140].

10.1.2 Node Search in Digraphs

Node-search games naturally extend to digraphs. However, in the translation another variation arises depending on how one views the constraints on the movement of the fugitive. One interpretation is that in the undirected case the fugitive moves along paths, so the natural translation would be to have the fugitive move along directed paths. Another view is that the fugitive moves to some other vertex in the same connected component, and here the natural translation would be to have the fugitive move within the same strongly connected component.[1] Both interpretations have been studied in the literature, the former giving characterizations of parameters such as DAG-width [18, 176] and directed path-width [16] and the latter giving a characterization of directed tree-width [129].

We define a variant of the node-search game in which only the most recently placed searchers may be removed; that is, the searchers must move in a last-in-first-out (LIFO) manner and we show that the minimum number of searchers required to capture a fugitive on a (di)graph with a LIFO-search is independent of:

- Whether the fugitive is visible or invisible,
- Whether the searchers use a monotone or non-monotone search, and
- Whether the fugitive is restricted to moving in searcher-free strongly connected components or along searcher-free directed paths.

This result is somewhat surprising: in the standard node-search game these options give rise to quite different parameters [16, 18, 140].

[1] A digraph G is *strongly connected* if for every two vertices $u, v \in V(G)$ it contains a path from u to v and a path from v to u.

We show that on digraphs the LIFO-search game characterizes a pre-existing measure, cycle-rank – one of the possible generalizations of tree-depth to digraphs (though as the definition of cycle-rank predates tree-depth by several decades, it is perhaps more correct to say that tree-depth is an analogue of cycle-rank on undirected graphs).

The cycle-rank of a digraph is an important parameter relating digraph complexity to other areas such as regular language complexity and asymmetric matrix factorization. It was defined by Eggan in [71], where it was shown to be a critical parameter for determining the star-height of regular languages. The success of tree-depth [72, 96, 110] rekindled interest in it as an important digraph parameter, especially from an algorithmic perspective.

It is well known that tree-depth can be characterized by a node-search game where a visible fugitive plays against searchers that are only placed and never moved [96]. In that paper, Ganian et al. considered one extension of this game to digraphs. Here we consider another natural extension, where the visible fugitive moves in strongly connected sets, and show that it also characterizes cycle-rank. From the above, we also obtain that the LIFO-search parameter is equivalent to the one of tree-depth.

Our final result uses these graph searching characterizations to define a dual parameter that characterizes structural obstructions for cycle-rank. We consider two kinds of obstructions. The first one is obtained from defining the notion of directed shelters. The second one is motivated by the havens of [129]. Both the directed shelters and LIFO-havens define simplified strategies for the fugitive. The game characterization then implies that these structural features are necessarily present when the cycle-rank of a graph is large. By showing that the aforementioned simplified strategies are also sufficient for the fugitive, we obtain a rare instance of an exact min-max theorem relating digraph parameters. This also implies that the notion of shelters when transferred to simple graphs characterizes structural obstructions for tree-depth.

The results appearing in this chapter can be summarized with the following characterizations of cycle-rank and tree-depth respectively.

Theorem 10.1. *Let G be a digraph, and k a positive integer. The following are equivalent:*

(i) *G has cycle-rank at most $k-1$.*

(ii) *On G, k searchers can capture a fugitive with a LIFO-search strategy.*

(iii) *On G, k searchers can capture a visible fugitive restricted to moving in strongly connected sets with a searcher-stationary search strategy.*

(iv) *G has no LIFO-haven of order greater than k.*

(v) *G has no directed shelter of thickness greater than k.*

Theorem 10.2. *Let G be a non-empty graph and k be a positive integer. Then the following are equivalent.*

(i) *G has tree-depth at most k.*

(ii) *On G, k searchers can capture an invisible and agile fugitive with a monotone LIFO-search strategy.*

(iii) *On G, k searchers can capture an invisible and agile fugitive with a LIFO-search strategy.*

(iv) *Every shelter in G has thickness at most k.*

(v) *On G, k searchers can capture a visible and agile fugitive with a monotone LIFO-search strategy.*

This chapter is organised as follows. In Section 10.2, we define the LIFO-search and searcher-stationary games and show that they characterize cycle-rank. In Section 10.3, we prove the min-max theorem for cycle-rank. In Section 10.4, we consider simple graphs and argue that our results imply the existense of a min-max theorem for LIFO-search and that the LIFO-search parameter is equivalent to the one of tree-depth.

10.2 Searching Games for Cycle-rank

Definition 10.1 (Cycle-rank)**.** The *cycle-rank* of a digraph G, $\mathbf{cr}(G)$, is defined as follows:

- If G is acyclic then $\mathbf{cr}(G) = 0$.
- If G is strongly connected then $\mathbf{cr}(G) = 1 + \min_{v \in V(G)} \mathbf{cr}(G \setminus \{v\})$.
- Otherwise $\mathbf{cr}(G) = \max_H \mathbf{cr}(H)$ where the maximum is taken over all strongly connected components H of G.

We begin by formally defining the LIFO-search game, and its variants, for digraphs. Each variation of the LIFO-search game gives rise to a digraph parameter corresponding to the minimum number of searchers required to capture the fugitive under the given restrictions. The main result of this section is that for any digraph all these parameters are equal. Furthermore, we show they are all equal to one more than the cycle-rank of the digraph.

10.2.1 LIFO-search on Digraphs

In summary, for the graph searching game in which we are interested the fugitive can run along searcher-free directed paths of any length, the searchers can move to any vertex in the graph, and the fugitive moves whilst the searchers are relocating. In this game, as in the classical node search game, the searchers first announce their move. Then the fugitive moves taking into account this information and finally the searchers carry their already announced move. The searcher's strategy may apply two types of moves in each step: either placement of a searcher on a vertex or removal of a searcher from a vertex with the restriction that only the most recently placed searchers may be removed. If a searcher is placed on the fugitive then he/she is captured and the searchers win, otherwise the fugitive wins. The goal is to determine the minimum number of searchers required to capture the fugitive. The variants we are primarily interested in are whether the fugitive is visible or invisible, and whether or not the fugitive must stay within the same strongly connected component when he/she is moving. As our fundamental definitions are dependent on these two options, we define four *game variants*: $\mathtt{i}, \mathtt{isc}, \mathtt{v}, \mathtt{vsc}$, corresponding to the visibility of the fugitive and whether he/she is constrained to moving within strongly connected components, that is, `i` and `v` correspond to an invisible and a visible fugitive respectively while `isc` and `vsc` correspond to an invisible and visible fugitive as above with the addition that the fugitives move inside the same strongly connected component. Then we parameterize our definitions by these variants.

Let us fix a digraph G. A *position* in a LIFO-search on G is a pair (X, R) where $X \in V(G)^*$ and R is a (possibly empty) induced subgraph of $G \setminus \{\!|X|\!\}$, where $\{\!|X|\!\}$ denotes the set of vertices in X. Intuitively X represents the position and ordered placement of the searchers and R

represents the part of G that the fugitive can reach (in the visible case) or the set of vertices where he/she might possibly be located (in the invisible case). We say that a position (X, R) is

- an $\mathtt{i}$-*position* if there are no edges in G from R to $G \setminus R$,
- an $\mathtt{isc}$-*position* if it is a union of strongly connected components of $G \setminus \{\!|X|\!\}$,
- a $\mathtt{v}$-*position* if there are no edges in G from R to $G \setminus R$ and $G[R]$ has a strongly connected component C with no edges from $G \setminus C$ to C, and
- a $\mathtt{vsc}$-*position* if R is a strongly connected component of $G \setminus \{\!|X|\!\}$.

To reflect how the game transitions to a new position during a round of the game we say, for $\mathbf{gv} \in \{\mathtt{i}, \mathtt{isc}, \mathtt{v}, \mathtt{vsc}\}$, a $\mathbf{gv}$-position (X', R') is a $\mathbf{gv}$-*successor* of (X, R) if either $X \preceq X'$ or $X' \preceq X$, with $|\{\!|X|\!\}\Delta\{\!|X'|\!\}| = 1$, and

- (for $\mathbf{gv} \in \{\mathtt{i}, \mathtt{v}\}$) For every $v' \in V(R')$ there is a $v \in V(R)$ and a directed path in $G \setminus (\{\!|X|\!\} \cap \{\!|X'|\!\})$ from v to v', or
- (for $\mathbf{gv} \in \{\mathtt{isc}, \mathtt{vsc}\}$) For every $v' \in V(R')$ there is a $v \in V(R)$ such that v and v' are contained in the same strongly connected component of $G \setminus (\{\!|X|\!\} \cap \{\!|X'|\!\})$.

Ideally we would like to assume games start from (ϵ, G). However, in the visible variants of the game this might not be a legitimate position. Thus, for $\mathbf{gv} \in \{\mathtt{v}, \mathtt{vsc}\}$, if (ϵ, G) is not a $\mathbf{gv}$-position we include it as a special case, and set as its $\mathbf{gv}$-successors all $\mathbf{gv}$-positions of the form (ϵ, R). We observe that in all variants, the successor relation is monotone in the sense that if (X, R) and (X, S) are positions with $S \subseteq R$ and (X', S') is a successor of (X, S), then there is a successor (X', R') of (X, R) with $S' \subseteq R'$.

For $\mathbf{gv} \in \{\mathtt{i}, \mathtt{isc}, \mathtt{v}, \mathtt{vsc}\}$, a ($\mathbf{gv}$-*LIFO-*)*search* in a digraph G from $\mathbf{gv}$-position (X, R) is a (finite or infinite) sequence of $\mathbf{gv}$-positions $(X, R) = (X_0, R_0), (X_1, R_1), \ldots$ where for all $i \geq 0$, (X_{i+1}, R_{i+1}) is a $\mathbf{gv}$-successor of (X_i, R_i). A LIFO-search is *complete* if either $R_n = \emptyset$ for some n, or it is infinite. We observe that if $R_n = \emptyset$, then $R_{n'} = \emptyset$ for all $n' \geq n$.

We say that a complete LIFO-search is *winning for the searchers* if $R_n = \emptyset$ for some n, otherwise it is *winning for the fugitive.* A complete LIFO-search from (ϵ, G)

- is *monotone* if $R_{i+1} \subseteq R_i$ for all i, that is, the fugitive does not occupy positions of the graph from which he/she has already been banned,
- is *searcher-stationary* if $X_i \preceq X_{i+1}$ for all i where $R_i \neq \emptyset$, and
- *uses at most k searchers* if $|X_i| \leq k$ for all i.

Whilst a complete LIFO-search from (ϵ, G) describes a single run of the game, we are more interested in the cases where one of the players (particularly the searchers) can always force a win, no matter what the other player chooses to do. For this, we introduce the notion of a strategy. For $\mathbf{gv} \in \{\mathtt{i}, \mathtt{isc}, \mathtt{v}, \mathtt{vsc}\}$, a (*searcher*) $\mathbf{gv}$-*strategy* is a (partial[2]) function σ from the set of all $\mathbf{gv}$-positions to $V(G)^*$ such that for all (X, R), $\sigma(X, R)$ is the first component of a $\mathbf{gv}$-successor of (X, R); so with the possible exception of $(X, R) = (\epsilon, G)$, either $\sigma(X, R) \preceq X$ or $X \preceq \sigma(X, R)$. A

[2] A strategy need only be defined for all positions (X, R) that can be reached from (ϵ, G) in a LIFO-search consistent with the strategy. However, as this definition is somewhat circular, we assume strategies are total.

gv-LIFO-search $(X_0, R_0), (X_1, R_1), \ldots$ is *consistent* with a **gv**-strategy σ if $X_{i+1} = \sigma(X_i, R_i)$ for all $i \geq 0$. A strategy σ is winning from (X, R) if all complete LIFO-searches from (X, R) consistent with σ are winning for the searchers. Likewise, a strategy is monotone (searcher-stationary, uses at most k searchers) if all consistent complete LIFO-searches from (ϵ, G) are monotone (searcher-stationary, use at most k searchers respectively). We say k searchers can capture a fugitive on G in the **gv**-game with a (monotone) LIFO-search strategy if there is a (monotone) **gv**-strategy that uses at most k searchers and is winning for the searchers from (ϵ, G).

For **gv** $\in \{\mathtt{i}, \mathtt{isc}, \mathtt{v}, \mathtt{vsc}\}$, we define the (*monotone*) `i`-*LIFO-search number* of G, $\mathrm{LIFO}^{\mathbf{gv}}(G)$ ($\mathrm{LIFO}^{\mathtt{m}\mathbf{gv}}(G)$), as the minimum k for which there is a (monotone) winning **gv**-strategy that uses at most k searchers. We also define the *visible, strongly connected, searcher-stationary search number* of G, $\mathrm{SS}^{\mathtt{vsc}}(G)$, as the minimum k for which there is a searcher-stationary winning `vsc`-strategy that uses at most k searchers.

In Section 10.3, we will also consider *fugitive* **gv**-*strategies*: a partial function ρ from $V(G)^* \times \mathcal{P}(G) \times V(G)^*$ to induced subgraphs of G, defined for (X, R, X') if (X, R) is a **gv**-position and X' is the first component of a **gv**-successor of (X, R). A LIFO-search $(X_0, R_0), (X_1, R_1), \ldots$ is *consistent* with a fugitive **gv**-strategy ρ if $R_{i+1} = \rho(X_i, R_i, X_{i+1})$ for all $i \geq 0$, and a fugitive strategy is *winning* if all consistent complete LIFO-searches are winning for the fugitive. In this section, a strategy will always refer to a searcher strategy.

10.2.2 Relating the Digraph Searching Parameters

We observe that in all game variants, a strategy that is winning from (X, R) can be used to define a strategy that is winning from (X, R') for any $R' \subseteq R$: the searchers can play as if the fugitive is located in the larger space; and from the monotonicity of the successor relation, the assumption that the actual set of locations of the fugitive is a subset of the assumed set of locations remains invariant. One consequence is that a winning strategy on G defines a winning strategy on any subgraph of G, so the search numbers we have defined are monotone with respect to the subgraph relation.

Proposition 10.1. *Let G be a digraph and G' a subgraph of G. Then:*

- $SS^{\mathtt{vsc}}(G') \leq SS^{\mathtt{vsc}}(G)$, *and*
- $LIFO^{\mathbf{gv}}(G') \leq LIFO^{\mathbf{gv}}(G)$ *for*

$$\mathbf{gv} \in \{\mathtt{i}, \mathtt{isc}, \mathtt{v}, \mathtt{vsc}, \mathtt{mi}, \mathtt{misc}, \mathtt{mv}, \mathtt{mvsc}\}.$$

Another consequence is that a winning strategy in the invisible fugitive variant defines a winning strategy when the fugitive is visible; and a winning strategy when the fugitive is not constrained to moving within strongly connected components defines a winning strategy when he/she is. This corresponds to our intuition of the fugitive being more (or less) restricted. Also, in all game variants, a monotone winning strategy is clearly a winning strategy, and because a searcher-stationary LIFO-search is monotone, a winning searcher-stationary strategy is a monotone winning strategy. These observations yield several inequalities between the search numbers defined above. For example $\mathrm{LIFO}^{\mathtt{vsc}}(G) \leq \mathrm{LIFO}^{\mathtt{mi}}(G)$ as any winning monotone $\mathtt{i}$-strategy is also a winning $\mathtt{vsc}$-strategy. The full set of these relations is shown in a Hasse diagram in Figure 10.1, with the larger measures towards the top.

The main result of this section is that all these digraph parameters are equal to one more than cycle-rank.

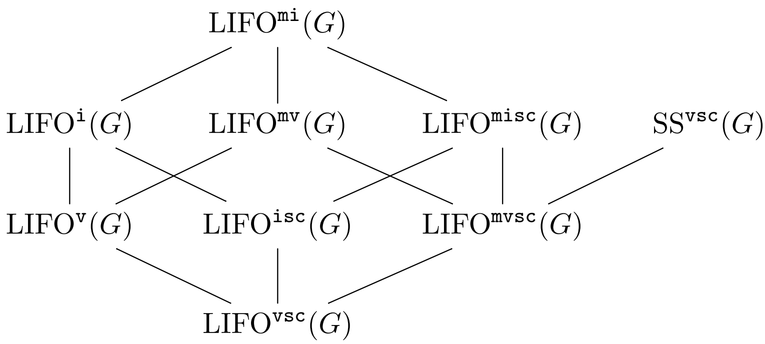


Figure 10.1: Trivial relations between digraph searching parameters.

Theorem 10.3. *For any digraph G:*

$$\begin{aligned}1+\mathbf{cr}(G) &= LIFO^{\mathtt{mi}}(G) = LIFO^{\mathtt{i}}(G) = LIFO^{\mathtt{misc}}(G) = LIFO^{\mathtt{isc}}(G)\\ &= LIFO^{\mathtt{mv}}(G) = LIFO^{\mathtt{v}}(G) = LIFO^{\mathtt{mvsc}}(G) = LIFO^{\mathtt{vsc}}(G)\\ &= SS^{\mathtt{vsc}}(G).\end{aligned}$$

Proof. From the above observations, to prove Theorem 10.3 it is sufficient to prove the following three inequalities:

(1) $\mathrm{LIFO}^{\mathtt{vsc}}(G) \geq \mathrm{SS}^{\mathtt{vsc}}(G)$,

(2) $\mathrm{SS}^{\mathtt{vsc}}(G) \geq 1 + \mathbf{cr}(G)$, and

(3) $1 + \mathbf{cr}(G) \geq \mathrm{LIFO}^{\mathtt{mi}}(G)$.

□

These are established with the following series of lemmata.

Lemma 10.1. *For any digraph G, $LIFO^{\mathtt{vsc}}(G) \geq SS^{\mathtt{vsc}}(G)$.*

Proof. We show that if a `vsc`-strategy is not searcher-stationary then it is not a winning strategy from (ϵ, G). The result then follows as this implies every winning $\mathtt{vsc}$-strategy is searcher-stationary. Let σ be a $\mathtt{vsc}$-strategy, and suppose $(X_0, R_0), (X_1, R_1), \ldots$ is a complete $\mathtt{vsc}$-LIFO-search from $(X_0, R_0) = (\epsilon, G)$ consistent with σ which is not searcher-stationary. Let j be the least index such that $X_j \succeq X_{j+1}$ and $R_j \neq \emptyset$. As $X_0 = \epsilon$, there exists $i < j$ such that $X_i = X_{j+1}$. By the minimality of j, and the assumption that we only place or remove one searcher in each round, $i = j - 1$. As $X_{j-1} \preceq X_j$, $R_j \subseteq R_{j-1}$, and as $X_{j+1} \preceq X_j$, $R_j \subseteq R_{j+1}$. As $R_j \neq \emptyset$, it follows that R_{j-1} and R_{j+1} are the same strongly connected component of $G \setminus \{\!|X_{j-1}|\!\}$. Thus (X_{j-1}, R_{j-1}) is a $\mathtt{vsc}$-successor of (X_j, R_j). As $\sigma(X_j, R_j) = X_{j+1} = X_{j-1}$, it follows that (X_0, R_0), (X_1, R_1),..., (X_{j-1}, R_{j-1}), (X_j, R_j), (X_{j-1}, R_{j-1}), (X_j, R_j), $\ldots$ is an infinite, and hence complete, $\mathtt{vsc}$-LIFO-search (from (ϵ, G)) consistent with σ. As $R_i \neq \emptyset$ for all $i \geq 0$, the LIFO-search is not winning for the searchers. Thus σ is not a winning strategy. □

Lemma 10.2. *For any digraph G, $SS^{\mathtt{vsc}}(G) \geq 1 + \mathbf{cr}(G)$.*

Proof. We prove this by induction on $|V(G)|$. If $|V(G)| = 1$, then $\mathrm{SS}^{\mathtt{vsc}}(G) = 1 = 1 + \mathbf{cr}(G)$.

Now suppose $\mathrm{SS}^{\mathtt{vsc}}(G') \geq 1 + \mathbf{cr}(G')$ for all digraphs G' with $|V(G')| < |V(G)|$. We first consider the case when G is not strongly connected. From Proposition 10.1, $\mathrm{SS}^{\mathtt{vsc}}(G) \geq \max_H \mathrm{SS}^{\mathtt{vsc}}(H)$, where the maximum is taken over all strongly connected components H of G. As G is not strongly connected, $|V(H)| < |V(G)|$ for all strongly connected components H of G. Therefore, by the induction hypothesis,

$$\begin{aligned}
\mathrm{SS}^{\mathtt{vsc}}(G) &\geq \max_H \mathrm{SS}^{\mathtt{vsc}}(H) \\
&\geq \max_H \left(1 + \mathbf{cr}(H)\right) \\
&= 1 + \mathbf{cr}(G).
\end{aligned}$$

Now suppose G is strongly connected. Let σ be a winning searcher-stationary $\texttt{vsc}$-strategy which uses $\mathrm{SS}^{\texttt{vsc}}(G)$ searchers. As (ϵ, G) is a legitimate $\texttt{vsc}$-position, if (X, R) is a $\texttt{vsc}$-successor of (ϵ, G) then $|X| = 1$. Thus $|\sigma(\epsilon, G)| = 1$. Let $\sigma(\epsilon, G) = v_0$. As σ is a searcher-stationary strategy which uses the minimal number of searchers, it follows that $\mathrm{SS}^{\texttt{vsc}}(G \setminus \{v_0\}) = \mathrm{SS}^{\texttt{vsc}}(G) - 1$. Thus, by the induction hypothesis,

$$\begin{aligned}
\mathrm{SS}^{\texttt{vsc}}(G) &= \mathrm{SS}^{\texttt{vsc}}(G \setminus \{v_0\}) + 1 \\
&\geq (1 + \mathbf{cr}(G \setminus \{v_0\})) + 1 \\
&\geq (1 + \min_{v \in V(G)} \mathbf{cr}(G \setminus \{v\})) + 1 \\
&= 1 + \mathbf{cr}(G).
\end{aligned}$$

□

Lemma 10.3. *For any digraph G, $1 + \mathbf{cr}(G) \geq \mathit{LIFO}^{\texttt{mi}}(G)$.*

Proof. We also prove this by induction on $|V(G)|$. If $|V(G)| = 1$, then $1 + \mathbf{cr}(G) = 1 = \mathrm{LIFO}^{\texttt{mi}}(G)$.

Now suppose $1 + \mathbf{cr}(G') \geq \mathrm{LIFO}^{\texttt{mi}}(G')$ for all digraphs G' with $|V(G')| < |V(G)|$. First we consider the case when G is not strongly connected. As $|V(H)| < |V(G)|$ for each strongly connected component H, by the inductive hypothesis, there is a monotone $\texttt{i}$-strategy, σ_H, which captures a fugitive using at most $1 + \mathbf{cr}(H)$ searchers. From the definition of cycle-rank, for each strongly connected component H of G, $\mathbf{cr}(G) \geq \mathbf{cr}(H)$, thus σ_H uses at most $1 + \mathbf{cr}(G)$ searchers. We define a monotone $\texttt{i}$-strategy which captures a fugitive on G with at most $1+\mathbf{cr}(G)$ searchers as follows. Intuitively, we search the strongly connected components of G in topological order using the monotone strategies σ_H. More precisely, let $H_1, H_2, \ldots, H_n$ be an ordering of the strongly connected components of G such that if there is an edge from H_i to H_j then $i < j$. We define σ as follows.

- $\sigma(\epsilon, G) = \sigma_{H_1}(\epsilon, H_1)$,
- For $1 \leq i$, if $\{\!|X|\!\} \subseteq H_i$ and $R = R' \cup \bigcup_{j=i+1}^{n} H_j$ where $\emptyset \neq R' \subseteq H_i$, $\sigma(X, R) = \sigma_{H_i}(X, R')$,
- For $1 \leq i < n$, if $\emptyset \neq \{\!|X|\!\} \subseteq H_i$ and $R = \bigcup_{j=i+1}^{n} H_j$ then $\sigma(X, R) = X'$ where X' is the maximal proper prefix of X.

From the definition of `i`-successors and the ordering of the strongly connected components if $(X_0, R_0), \dots (X_n, R_n)$ is an `i`-LIFO-search on G where $\{\!|X_n|\!\} \subseteq H_i$ and $\bigcup_{j>i} H_j \subseteq R_{n-1} \subseteq \bigcup_{j \geq i} H_j$, then $\bigcup_{j>i} H_j \subseteq R_n \subseteq \bigcup_{j \geq i} H_j$. It follows (by induction on the length of a LIFO-search) that every LIFO-search from (ϵ, G) consistent with σ can be divided into a sequence of LIFO-searches $\lambda_1, \lambda_2, \dots, \lambda_n$, where λ_i can be viewed as a LIFO-search consistent with σ_{H_i} with $\bigcup_{j>i} H_j$ added to the second component of every position. Thus if each σ_{H_i} is monotone, winning and uses at most $1 + \mathbf{cr}(G)$ searchers, then σ is also monotone, winning and uses at most $1 + \mathbf{cr}(G)$ searchers.

Now suppose G is strongly connected. Let v_0 be the vertex which minimizes $f(v) = \mathbf{cr}(G \setminus \{v\})$. Let $G' = G \setminus \{v_0\}$, so $\mathbf{cr}(G) = 1 + \mathbf{cr}(G')$. By the induction hypothesis, there exists a winning monotone `i`-strategy σ' which uses at most $1 + \mathbf{cr}(G')$ searchers to capture a fugitive on G'. We define an `i`-strategy σ on G which uses at most $2 + \mathbf{cr}(G') = 1 + \mathbf{cr}(G)$ searchers as follows. Initially, place (and keep) a searcher on v_0, then play the strategy σ' on $G \setminus \{v_0\}$. More precisely, $\sigma(\epsilon, G) = v_0$ and $\sigma(v_0 X, R) = v_0 \cdot \sigma'(X, R)$. Clearly any LIFO-search consistent with σ can be viewed as a LIFO-search consistent with σ' prepended with the position (ϵ, G) and where the first component of every position is prepended with v_0. Thus if σ' is monotone, then σ is monotone, and if σ' is winning then σ is winning. Thus σ is a monotone winning $\mathtt{i}$-strategy which uses at most $1 + \mathbf{cr}(G)$ searchers. □

10.2.3 Relation with Other Graph Parameters

With a characterization of cycle-rank in terms of several graph searching games we can compare it with other digraph measures defined by similar games. In particular, the directed path-width of a digraph, $\mathbf{dpw}(G)$, which can be characterized by an invisible-fugitive graph searching game [16], and the DAG-depth, $\mathbf{dd}(G)$ which can be characterized by a visible-fugitive, searcher-stationary searching game [96]. Whilst the relations we present here are known [96, 110], using the game characterizations we obtain a more simple and more intuitive proof.

Corollary 10.1. *For any digraph G,*

$$\mathbf{dpw}(G) \leq \mathbf{cr}(G) \leq \mathbf{dd}(G) - 1.$$

10.3 Obstructions for Cycle-rank

In this section, we consider the dual parameter arising from considering the graph searching games from the fugitive's perspective. We show that it can be characterized by two types of structural features, akin to the havens and brambles used to dually characterize tree-width [217]. To do so we first define the notion of the directed shelter of a digraph, a structural obstruction which we show to be dual to cycle-rank.

Definition 10.2. A *directed shelter* of a digraph G is a collection $\mathcal{S}$ of non-empty strongly connected sets of vertices such that for any non-minimal

$S \in \mathcal{S}$

$$\bigcap\{S' : S' \in M_{\mathcal{S}}(S)\} = \emptyset,$$

where $M_{\mathcal{S}}(S)$ is the $\subseteq$-maximal elements of $\{S' \in \mathcal{S} : S' \subset S\}$. The *thickness* of a shelter $\mathcal{S}$ is the minimal length of a maximal $\subseteq$-chain.

The second structural obstruction we consider is motivated by the definition of a haven in [129], a structural feature dual to directed tree-width.

Definition 10.3. A *LIFO-haven of order k* of a digraph G is a function h from $V(G)^{<k}$ to induced subgraphs of G such that:

(H1) $h(X)$ is a non-empty strongly connected component of $G \setminus \{\!|X|\!\}$, and

(H2) If $X \preceq Y$ and $|Y| < k$ then $h(Y) \subseteq h(X)$.

Whilst Adler [2] has shown that the havens of [129] do not give an exact min-max characterization of directed tree-width and Safari [214] has shown that directed versions of havens and brambles give rise to distinct parameters, we show that LIFO-havens and directed shelters both give a tight min-max characterization of cycle-rank.

Theorem 10.4 (Min-max Theorem for Cycle-rank)**.** *Let G be a digraph and k a positive integer. The following are equivalent:*

(i) G has cycle-rank less than k.

(ii) G has no LIFO-haven of order greater than k.

(iii) G has no directed shelter of thickness greater than k.

Proof. $(i) \Rightarrow (ii)$. Assume that it is not the case that G has no LIFO-haven of order greater than k, that is, G has a LIFO-haven h of order at least $k+1$. We show that the fugitive has a winning strategy against k searchers, so by Theorem 10.3, $\mathbf{cr}(G) \geq k$. Define a $\mathtt{vsc}$-strategy ρ for the fugitive (against k searchers) by defining $\rho(X, R, X') = h(X')$ for all suitable triples (X, R, X'). From (H1), $(X', \rho(X, R, X'))$ is a valid $\mathtt{vsc}$-position. Furthermore, (H2) implies that if (X, R) is a $\mathtt{vsc}$-position such that $R = h(X)$, then $(X', \rho(X, R, X'))$ is a $\mathtt{vsc}$-successor of (X, R), so ρ is a $\mathtt{vsc}$-strategy (defined for all LIFO-searches that use at most k searchers). Also, if $(X_0, R_0), (X_1, R_1) \ldots$ is a complete LIFO-search consistent with ρ then $R_i = h(X_i)$ for all $i > 0$. As $h(X) \neq \emptyset$ when $|X| \leq k$, it follows that all consistent complete LIFO-searches that use at most k searchers are winning for the fugitive. Thus ρ is a winning strategy for the fugitive, so $\mathrm{LIFO}^{\mathtt{vsc}}(G) > k$. By Theorem 10.3, $\mathbf{cr}(G) \geq k$.

$(ii) \Rightarrow (iii)$. We show that a directed shelter $\mathcal{S}$ of thickness at least k can be used to define a haven of order at least k. For each $X \in V(G)^{<k}$ we define $S_X \in \mathcal{S}$ inductively as follows. For $X = \epsilon$, let S_ϵ be any $\subseteq$-maximal element of $\mathcal{S}$. Note that $\{S \in \mathcal{S} \mid S \subset S_\epsilon\}$ is a directed shelter of thickness at least $k-1$. Now suppose $X = X'v$, $S_{X'}$ is defined, $S_{X'} \cap \{\!|X'|\!\} = \emptyset$, and $\mathcal{S}_{X'} = \{S \in \mathcal{S} \mid S \subset S_{X'}\}$ is a directed shelter of thickness at least $k - 1 - |X'|$. From the definition of a directed shelter, there exists a $\subseteq$-maximal element of $\mathcal{S}_{X'}$ that does not contain v, as otherwise $v \in S$ for all $S \in M_{\mathcal{S}}(S_{X'})$. Let S_X be that element. As $S_{X'} \cap \{\!|X'|\!\} = \emptyset$ and $v \notin S_X$, it follows that $S_X \cap \{\!|X|\!\} = \emptyset$. Further, $\{S \in \mathcal{S} \mid S \subset S_X\}$ is a directed shelter of thickness at least $(k - 1 - |X'|) - 1 = k - 1 - |X|$, satisfying the assumptions necessary for the next stage of the induction. Now for all $X \in V(G)^{<k}$, S_X is a non-empty strongly connected set such that $S_X \cap \{\!|X|\!\} = \emptyset$. Thus there is a unique strongly connected component of

$G \setminus \{\!|X|\!\}$ that contains S_X. Defining $h(X)$ to be that component we see that h satisfies (H1). For (H2), from the definition of S_X, if $X \preceq Y$ and $|Y| < k$, then $S_X \supseteq S_Y$, so $h(X) \supseteq h(Y)$. Therefore h is a haven of order at least k.

(iii) $\Rightarrow$ *(i)*. Again, we prove the contrapositive. Suppose $\mathbf{cr}(G) \geq k$. Let G' be a strongly connected component of G which has cycle-rank at least k. We prove by induction on k that G', and hence G, has a directed shelter of thickness at least $k+1$. Every digraph with $|V(G)| \geq 1$ has a directed shelter of thickness 1: take $\mathcal{S} = \{\{v\}\}$ for some $v \in V(G)$. Thus for $k = 0$, the result is trivial. Now suppose for $k' < k$ every digraph of cycle-rank at least k' contains a directed shelter of thickness at least $k'+1$. For $v \in V(G')$, let $G'_v = G' \setminus \{v\}$. From the definition of cycle-rank, $\mathbf{cr}(G'_v) \geq k-1$ for all $v \in V(G')$. Thus, by the induction hypothesis, G'_v contains a directed shelter, $\mathcal{S}_v$, of thickness at least $(k-1)+1$. As $v \notin S$ for all $S \in \mathcal{S}_v$, it follows that $\mathcal{S} = \{G'\} \cup \bigcup_{v \in V(G')} \mathcal{S}_v$ is a directed shelter of G. As $\mathcal{S}_v$ has thickness at least k for all $v \in V(G')$, $\mathcal{S}$ has thickness at least $k+1$. $\square$

Combining Theorems 10.3 and 10.4 we obtain the following.

Theorem 10.5. *Let G be a digraph and k a positive integer. The following are equivalent:*

(i) *G has cycle-rank at most $k-1$.*

(ii) *On G, k searchers can capture a fugitive with a LIFO-search strategy.*

(iii) *On G, k searchers can capture a visible fugitive restricted to moving in strongly connected sets with a searcher-stationary search strategy.*

(iv) *G has no LIFO-haven of order greater than k.*

(v) G has no directed shelter of thickness greater than k.

10.4 LIFO-search in Simple Graphs

In this section, we consider the consequences of our results to simple graphs. Given a graph G, we define the digraph G_d where $V(G_d) = V(G)$ and $E(G_d) = \{(x, y) \mid \{x, y\} \in E(G)\}$, that is, G_d is obtained from G after replacing every edge $\{x, y\} \in E(G)$ with the arcs (x, y) and (y, x). From Definitions 10.1 and 4.1 we get that.

Observation 10.1. *For every graph G, $\mathbf{td}(G) = \mathbf{cr}(G_d) + 1$.*

We now give the definition of shelters when restricted to simple graphs.

Definition 10.4 (Shelter)**.** A *shelter* of G is a collection $\mathcal{S}$ of non-empty connected sets in G such that for every non-minimal set $S \in \mathcal{S}$ no vertex belongs to all its children, in other words,

$$\bigcap\{S' \mid S' \in M_{\mathcal{S}}(S)\} = \emptyset,$$

where $M_{\mathcal{S}}(S)$ is the $\subseteq$-maximal elements of $\{S' \in \mathcal{S} : S' \subset S\}$. The *thickness* of a shelter $\mathcal{S}$ is the minimal length of a maximal $\subseteq$-chain.

In Figure 10.2 you can see a shelter of P_9 of thickness 4 and in Figure 10.3 you can see the steps of a monotone winning LIFO-strategy in P_9 using at most 4 searchers.

Observation 10.1 ensures that we may restate Theorem 10.5 for simple graphs in the following way.

Theorem 10.6. *Let G be a non-empty graph and k be a positive integer. Then the following are equivalent.*

(i) G has tree-depth at most k.

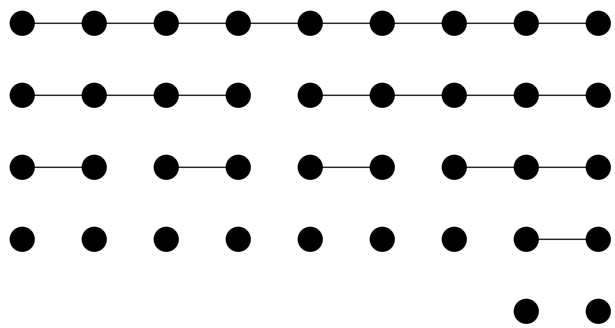


Figure 10.2: A shelter of P_9 that has thickness 4.

(ii) *On G, k searchers can capture an invisible and agile fugitive with a monotone LIFO-search strategy.*

(iii) *On G, k searchers can capture an invisible and agile fugitive with a LIFO-search strategy.*

(iv) *Every shelter in G has thickness at most k.*

(v) *On G, k searchers can capture a visible and agile fugitive with a monotone LIFO-search strategy.*

However, we wish to also include a simpler straightforward proof for the case of ordinary graphs.

Proof. $(i) \Rightarrow (ii)$: We proceed by induction on the quantity $k \cdot (|V(G)| - 2) + |\mathcal{C}(G)|$ where $k = \mathbf{td}(G)$. Notice that $k \cdot (|V(G)| - 2) + |\mathcal{C}(G)| \geq 0$, where the equality holds when $|V(G)| = 1$, and therefore $k = 1$. In such a case, the winning strategy is $\mathcal{R} = [\epsilon, x]$ where $\{x\} = V(G)$. Suppose now that the statement holds for all G where $k \cdot (|V(G)| - 2) + |\mathcal{C}(G)| < \rho$ and let G be a graph such that $k \cdot (|V(G)| - 2) + |\mathcal{C}(G)| = \rho$. Let

$\mathcal{C}(G) = \{G_1, \ldots, G_\sigma\}$ where $\sigma = |\mathcal{C}(G)|$. From the definition of tree-depth it follows that $k_i = \mathbf{td}(G_i) \leq k$, for $i \in \{1, \ldots, \sigma\}$. We distinguish two cases.

Case i. If $\sigma \geq 2$, then $k_i \cdot (|V(G_i)| - 2) + |\mathcal{C}(G_i)| \leq k \cdot (|V(G_i)| - 2) + 1 < k \cdot (|V(G_i)| - 2) + \sigma = \rho$, $i \in \{1, \ldots, \sigma\}$ and, from the induction hypothesis, there exists a monotone winning strategy $\mathcal{R}_i$ for G_i using at most k_i searchers. Then $\mathcal{R} = \mathcal{R}_1 \oplus \ldots \oplus \mathcal{R}_\sigma$ is a monotone winning strategy on G using at most k searchers (we use the term $\oplus$ to denote the concatenation of two sequences).

Case ii. If $\sigma = 1$, then from the definition of tree-depth, G contains a vertex x such that the graph $G^- = G \setminus x$ has tree-depth $k^- = k - 1$. As the tree-depth of G^- is a positive integer, it follows that $k \geq 2$. Notice now that $k^- \cdot (|V(G^-)| - 2) + |\mathcal{C}(G^-)| \leq k \cdot (|V(G)| - 2) - k + 2 < k \cdot (|V(G)| - 2) + 1 = k \cdot (|V(G)| - 2) + \sigma = \rho$. Therefore, we may apply the induction hypothesis on G^- and obtain a monotone winning strategy $\mathcal{R}^- = [w_1, \ldots, w_\rho]$ on G^- using at most $k - 1$ searchers. But then the strategy $\mathcal{R} = [\epsilon, xw_1, \ldots, xw_\rho]$ is a monotone winning strategy on G using at most k searchers.

$(iii) \Rightarrow (iv)$: We prove the contrapositive using induction on k. On other words, we show that for every $k \geq 0$, if $\mathcal{K}$ is a shelter of G that has thickness at least $k+1$, then there is no winning strategy on G using at most k searchers. For $k = 0$ this is obvious as we need at least one searcher to capture a fugitive escaping in a non-empty graph. Assume now that the statement is correct for all k where $0 \leq k < k'$. We will now prove that it also holds for $k = k'$.

Let $\mathcal{K}$ be a shelter of G of thickness at least $k + 1$ with minimum number of maximal sets. Notice that such a shelter should have only one maximal set, as for every maximal set T of $\mathcal{K}$, $\{T\} \cup \{S \mid$

S is a descendant of T in $\mathcal{K}\}$ is also a shelter of G of thickness at least $k+1$. Let X be the maximal set of $\mathcal{K}$. We assume that G is connected, otherwise we work on the connected component of G that contains the maximal set of $\mathcal{K}$. Let $\mathcal{R} = [w_0, \ldots, w_\rho]$ be a winning strategy on G using at most k searchers and of minimum length. We say that the i-th move of $\mathcal{R}$ is a *starting move* if $w_{i-1} = \epsilon$. Clearly $\mathcal{R}$ should have at least one starting move as the first placement move is one of them. Moreover, $\mathcal{R}$ cannot have any other starting move. Indeed, if the i-th move is such a move where $i > 1$, then $\mathcal{R}' = [w_{i-1}, w_i, \ldots, w_\rho]$ would also be a winning strategy using at most k searchers, contradicting the minimality of $\mathcal{R}$. This implies that the searcher placed first is never removed from the graph. Assume now that x is the vertex where the first placement move occurs. We claim that $G^- = G \setminus x$ contains a shelter $\mathcal{K}^-$ of thickness at least k. We distinguish two cases.

Case 1. If x intersects X then, from the definition of a shelter, x cannot belong to all the children of X. Let X^- be such a child. Notice that X^- is the maximal set of the shelter $\mathcal{K}^-$ defined by X^- and its descendants. It follows that $\mathcal{K}^-$ is a shelter of $G \setminus x$ of thickness at least k.

Case 2. If x does not intersect X, then $\mathcal{K} = \mathcal{K} \setminus x$ and thus $\mathcal{K}^- = \mathcal{K}$ is a shelter of $G^- = G \setminus x$ of thickness at least $k+1 \geq k$. Notice now that the strategy $\mathcal{R}^- = [w_1^-, \ldots, w_\rho^-]$, created if we remove x from w_i for each $i \in \{1, \ldots, \rho\}$, is a winning strategy on G^- using at most $k-1$ searchers, contradicting the induction hypothesis for $k = k' - 1$.

(iv) $\Rightarrow$ *(i)*: Again we prove the contrapositive, that is, for every $k \geq 1$, if $\mathbf{td}(G) \geq k$, then there exists a shelter of G that has thickness at least k. The case where $k = 1$ is obvious as every connected component of a non-empty graph is a one element shelter of thickness 1. Assume now that the statement is correct for all k where $0 \leq k < k'$. We will prove that it

also holds for $k = k'$. Let G be a graph where $\mathbf{td}(G) \geq k$ and let G' be one of its connected components that has tree-depth at least k. We prove that G' has a shelter $\mathcal{K}$ of thickness at least k. As $\mathcal{K}$ is also a shelter of G, G has a shelter of thickness k. As $\mathbf{td}(G') \geq k$ and G' is connected, it follows that for every $x \in V(G')$, $\mathbf{td}(G' \setminus x) \geq k - 1$. From the induction hypothesis, for every $x \in V(G')$, $G^{(x)} = G' \setminus x$ contains a shelter $\mathcal{K}^{(x)}$ of thickness at least $k - 1$. Notice that $\mathcal{K} = \{V(G')\} \cup \bigcup_{x \in V(G')} \mathcal{K}^{(x)}$ is a shelter of G', as $\bigcap_{x \in V(G')}(V(G') \setminus x) = \emptyset$. Moreover, $\mathcal{K}$ has thickness one more than the minimum thickness of $\mathcal{K}^{(x)}, x \in V(G')$, therefore $\mathcal{K}$ is a shelter of thickness at least k, as required.

Finally, it is trivial to see that $(ii) \Rightarrow (v)$ (since we can follow the same strategy in the case where the fugitive is visible), while for $(v) \Rightarrow (ii)$ it is enough to recall that from the results in [96] there is a searcher-stationery strategy for a visible and agile fugitive which is also a monotone LIFO-search strategy. $\square$

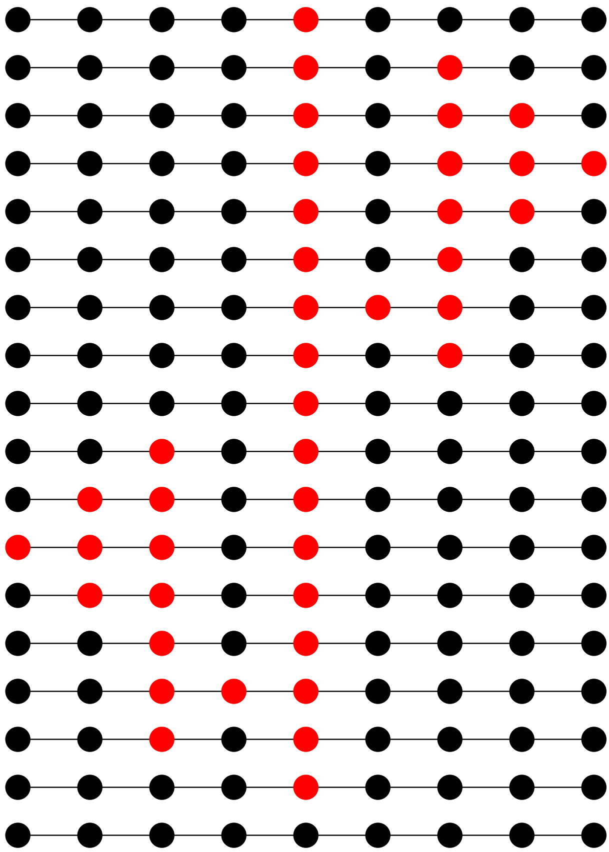


Figure 10.3: A monotone winning LIFO-strategy in P_9 using 4 searchers.

Bibliography

[1] Faisal N. Abu-Khzam and Michael A. Langston. Graph coloring and the immersion order. In Tandy Warnow and Binhai Zhu, editors, *COCOON*, volume 2697 of *Lecture Notes in Computer Science*, pages 394–403. Springer, 2003.

[2] Isolde Adler. Directed tree-width examples. *Journal of Combinatorial Theory, Series B*, 97(5):718–725, 2007.

[3] Isolde Adler, Arthur M. Farley, and Andrzej Proskurowski. Obstructions for linear rankwidth at most 1. *CoRR*, abs/1106.2533, 2011.

[4] Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In Shang-Hua Teng, editor, *SODA*, pages 641–650. SIAM, 2008.

[5] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Tight

bounds for linkages in planar graphs. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 110–121. Springer, 2011.

[6] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.

[7] Jochen Alber, Hans L. Bodlaender, Henning Fernau, and Rolf Niedermeier. Fixed parameter algorithms for planar dominating set and related problems. In *6th Scandinavian Workshop on Algorithm Theory—SWAT 2000 (Bergen)*, pages 97–110. Springer, Berlin, 2000.

[8] Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving max-r-sat above a tight lower bound. In *SODA*, pages 511–517, 2010.

[9] Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving max-r-sat above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011.

[10] Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast fast. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP (1)*, volume 5555 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009.

[11] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. Assoc. Comput. Mach.*, 42(4):844–856, 1995.

[12] Eyal Amir. Approximation algorithms for treewidth. *Algorithmica*, 56(4):448–479, 2010.

[13] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.

[14] Stefan Arnborg, Andrzej Proskurowski, and Derek G. Corneil. Forbidden minors characterization of partial 3-trees. *Discrete Mathematics*, 80(1):1–19, 1990.

[15] R. Balasubramanian, Michael Fellows, and Venkatesh Raman. An improved fixed-parameter algorithm for vertex cover. *Information Proccessing Letters*, 65:163–168, 1998.

[16] János Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.

[17] Rémy Belmonte, Pim van 't Hof, Marcin Kaminski, Daniël Paulusma, and Dimitrios M. Thilikos. Characterizing graphs of small carving-width. In Guohui Lin, editor, *COCOA*, volume 7402 of *Lecture Notes in Computer Science*, pages 360–370. Springer, 2012.

[18] Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. DAG-width and parity games. In *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science*, pages 524–536, 2006.

[19] Therese C. Biedl and Michael Kaufmann. Area-efficient static and incremental graph drawings. In Rainer E. Burkard and Gerhard J. Woeginger, editors, *ESA*, volume 1284 of *Lecture Notes in Computer Science*, pages 37–52. Springer, 1997.

[20] Daniel Bienstock and Paul D. Seymour. Monotonicity in graph searching. *Journal of Algorithms*, 12:239–245, 1991.

[21] Hans L. Bodlaender, Jitender S. Deogun, Klaus Jansen, Ton Kloks, Dieter Kratsch, Haiko Müller, and Zsolt Tuza. Rankings of graphs. *SIAM J. Discrete Math.*, 11(1):168–181 (electronic), 1998.

[22] Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors. *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*. Springer, 2012.

[23] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[24] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) Kernelization. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 629–638. IEEE, 2009.

[25] Hans L. Bodlaender and Dimitrios M. Thilikos. Graphs with branchwidth at most three. *Journal of Algorithms*, 32:167–197, 1999.

[26] Hans Leo Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21:358–402, 1996.

[27] B. Bollobás and A. D. Scott. Better bounds for Max Cut. In *Contem-*

porary combinatorics, volume 10 of *Bolyai Soc. Math. Stud.*, pages 185–246. János Bolyai Math. Soc., Budapest, 2002.

[28] Liming Cai and David Juedes. On the existence of subexponential parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):789 – 807, 2003. Parameterized Computation and Complexity 2003.

[29] Kevin Cattell, Michael J. Dinneen, Rodney G. Downey, Michael R. Fellows, and Michael A. Langston. On computing graph minor obstruction sets. *Theor. Comput. Sci.*, 233(1-2):107–127, 2000.

[30] L. Sunil Chandran and Fabrizio Grandoni. Refined memorization for vertex cover. *Inf. Process. Lett.*, 93(3):123–131, 2005.

[31] Moses Charikar, editor. *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*. SIAM, 2010.

[32] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.

[33] Jianer Chen and Songjian Lu. Improved algorithms for weighted and unweighted set splitting problems. In *COCOON*, volume 4598 of *Lecture Notes in Computer Science*, pages 537–547, 2007.

[34] Rajesh Hemant Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing fpt algorithms for cut problems using randomized contractions. *CoRR*, abs/1207.4079, 2012.

[35] Janka Chlebíková. The structure of obstructions to treewidth and pathwidth. *Discrete Applied Mathematics*, 120(1-3):61–71, 2002.

[36] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[37] Bruno Courcelle, Rodney G. Downey, and Michael R. Fellows. A note on the computability of graph minor obstruction sets for monadic second order ideals. In *Proceedings of the First Japan-New Zealand Workshop on Logic in Computer Science (Auckland, 1997)*, volume 3, pages 1194–1198 (electronic), 1997.

[38] Robert Crowston, Michael R. Fellows, Gregory Gutin, Mark Jones, Frances A. Rosamond, Stéphan Thomassé, and Anders Yeo. Simultaneously satisfying linear equations over F_2: Maxlin2 and max-r-lin2 parameterized above average. In *FSTTCS*, volume 13 of *LIPIcs*, pages 229–240, 2011.

[39] Robert Crowston, Gregory Gutin, Mark Jones, Eun Jung Kim, and Imre Z. Ruzsa. Systems of linear equations over $\mathbb{F}_2$ and problems parameterized above average. In *SWAT*, volume 6139 of *Lecture Notes in Computer Science*, pages 164–175, 2010.

[40] Robert Crowston, Gregory Gutin, Mark Jones, Venkatesh Raman, and Saket Saurabh. Parameterized complexity of maxsat above average. In *LATIN*, pages 184–194, 2012.

[41] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In Ostrovsky [178], pages 150–159.

[42] Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *LICS'07*, pages 270–279. IEEE Computer Society, 2007.

[43] Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Approximation schemes for first-order definable optimisation problems. In *LICS*, pages 411–420. IEEE Computer Society, 2006.

[44] Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009)*, pages 157–168, 2009.

[45] P. de la Torre, R. Greenlaw, and A. A. Schäffer. Optimal edge ranking of trees in polynomial time. *Algorithmica*, 13(6):592–618, 1995.

[46] Frank K. H. A. Dehne, Michael R. Fellows, and Frances A. Rosamond. An fpt algorithm for set splitting. In *WG*, volume 2880 of *Lecture Notes in Computer Science*, pages 180–191, 2003.

[47] Frank K. H. A. Dehne, Michael R. Fellows, Frances A. Rosamond, and Peter Shaw. Greedy localization, iterative compression, modeled crown reductions: New fpt techniques, an improved algorithm for set splitting, and a novel 2k kernelization for vertex cover. In *IWPEC*, volume 3162 of *Lecture Notes in Computer Science*, pages 271–280, 2004.

[48] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensional parameters and local treewidth. *SIAM J. Discrete Math.*, 18(3):501–511, 2004.

[49] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, 2005.

[50] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005.

[51] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. *J. Comput. Syst. Sci.*, 69(2):166–195, 2004.

[52] Erik D. Demaine and Mohammadtaghi Hajiaghayi. Bidimensionality: new connections between FPT algorithms and PTASs. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pages 590–601, New York, 2005. ACM-SIAM.

[53] Erik D. Demaine and MohammadTaghi Hajiaghayi. Linearity of grid minors in treewidth with applications through bidimensionality. *Combinatorica*, 28(1):19–36, 2008.

[54] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. *Algorithmica*, 41:245–267, 2005.

[55] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Dimitrios M.

Thilikos. The bidimensional theory of bounded-genus graphs. *SIAM J. Discrete Math.*, 20(2):357–371, 2006.

[56] Nick D. Dendris, Lefteris M. Kirousis, and Dimitrios M. Thilikos. Fugitive-search games on graphs and related parameters. *Theoretical Computer Science*, 172(1-2):233–254, 1997.

[57] J. S. Deogun, T. Kloks, D. Kratsch, and H. Müller. On vertex ranking for permutation and other graphs. In *STACS 94 (Caen, 1994)*, volume 775 of *Lecture Notes in Comput. Sci.*, pages 747–758. Springer, Berlin, 1994.

[58] M. DeVos, Z. Dvořák, J. Fox, J. McDonald, B. Mohar, and D. Scheide. Minimum degree condition forcing complete graph immersion. *ArXiv e-prints*, January 2011.

[59] Matt DeVos, Ken-Ichi Kawarabayashi, Bojan Mohar, and Haruko Okamura. Immersing small complete graphs. *Ars Math. Contemp.*, 3(2):139–146, 2010.

[60] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

[61] Michael J. Dinneen and Liu Xiong. Minor-order obstructions for the graphs of vertex cover 6. *Journal of Graph Theory*, 41(3):163–178, 2002.

[62] G. A. Dirac and S. Schuster. A theorem of Kuratowski. *Nederl. Akad. Wetensch. Proc. Ser. A.* **57** = *Indagationes Math.*, 16:343–348, 1954.

[63] G. A. Dirac and S. Schuster. Corrigendum: "A theorem of Kuratowski". *Nederl. Akad. Wetensch. Proc. Ser. A 64 = Indag. Math.*, 23:360, 1961.

[64] Frederic Dorn. Dynamic programming and fast matrix multiplication. In *Proceedings of the 14th Annual European Symposium on Algorithms (ESA 2006)*, volume 4168 of *Lecture Notes in Comput. Sci.*, pages 280–291. Springer, Berlin, 2006.

[65] Frederic Dorn. *Designing Subexponential Algorithms: Problems, Techniques & Structures.* PhD thesis, Universitetet i Bergen, 2007.

[66] Rodney G. Downey and Michael R. Fellows. *Parameterized complexity.* Monographs in Computer Science. Springer-Verlag, New York, 1999.

[67] Feodor F. Dragan, Fedor V. Fomin, and Petr A. Golovach. Spanners in sparse graphs. *J. Comput. Syst. Sci.*, 77(6):1108–1119, 2011.

[68] Zdenek Dvorak. Constant-factor approximation of domination number in sparse graphs. *CoRR*, abs/1110.5190, 2011.

[69] Zdenek Dvorak, Archontia C. Giannopoulou, and Dimitrios M. Thilikos. Forbidden graphs for tree-depth. *Eur. J. Comb.*, 33(5):969–979, 2012.

[70] C. S. Edwards. Some extremal properties of bipartite subgraphs. *Canad. J. Math.*, 25:475–485, 1973.

[71] L. C. Eggan. Transition graphs and the star-height of regular events. *Michigan Mathematical Journal*, 10(4):385–397, 1963.

[72] Stanley Eisenstat and Joseph Liu. The theory of elimination trees for sparse unsymmetric matrices. *SIAM Journal of Matrix Analysis & Applications*, 26(3):686–705, 2005.

[73] Herbert B. Enderton. *A mathematical introduction to logic.* Academic Press, 1972.

[74] P. Erdős, G.O.H. Katona, and Hongrie) Colloquium on graph theory (1966. Tihany. *Theory of Graphs: Proceedings of the Colloquium [on Graph Theory], Held at Tihany, Hungary, September 1966. Edited by P. Erdős and G. Katona.* Akadémiai Kiadó, 1968.

[75] P. Erdős. On some extremal problems in graph theory. *Israel J. Math.*, 3:113–116, 1965.

[76] Paul Erdős and Louis Pósa. On independent circuits contained in a graph. *Canad. J. Math.*, 17:347–352, 1965.

[77] Michael R. Fellows, Bart M.P. Jansen, and Frances Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, (0):–, 2012.

[78] Michael R. Fellows and Michael A. Langston. Nonconstructive tools for proving polynomial-time decidability. *J. Assoc. Comput. Mach.*, 35(3):727–739, 1988.

[79] Michael R. Fellows and Michael A. Langston. On search, decision, and the efficiency of polynomial-time algorithms. *J. Comput. System Sci.*, 49(3):769–779, 1994.

[80] Henning Fernau. A top-down approach to search-trees: Improved algorithmics for 3-hitting set. *Electronic Colloquium on Computational Complexity (ECCC)*, (073), 2004.

[81] Michael Ferrara, Ronald J. Gould, Gerard Tansey, and Thor Whalen. On h-immersions. *Journal of Graph Theory*, 57(3):245–254, 2008.

[82] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.

[83] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Contraction obstructions for treewidth. *J. Comb. Theory, Ser. B*, 101(5):302–314, 2011.

[84] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. Measure and conquer: Domination - a case study. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 191–203. Springer, 2005.

[85] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and kernelization. In Thomas Schwentick and Christoph Dürr, editors, *STACS*, volume 9 of *LIPIcs*, pages 189–200. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[86] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar f-deletion: Approximation and optimal fpt algorithms. *CoRR*, abs/1204.4230, 2012.

[87] Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Bidimensionality and EPTAS. In *22st ACM–SIAM Symposium on Discrete Algorithms (SODA 2011)*, 2011.

[88] Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Subexponential algorithms for partial cover problems. *Inf. Process. Lett.*, 111(16):814–818, 2011.

[89] Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In Rabani [180], pages 1563–1575.

[90] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on h-minor-free graphs. In Rabani [180], pages 82–93.

[91] Fedor V. Fomin, Saket Saurabh, and Dimitrios M. Thilikos. Strengthening erdös-pósa property for minor-closed graph classes. *Journal of Graph Theory*, 66(3):235–240, 2011.

[92] Fedor V. Fomin and Dimitrios M. Thilikos. Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004)*, volume 3142 of *Lecture Notes in Computer Science*, pages 581–592. Springer, 2004.

[93] András Frank, Tamás Király, and Matthias Kriesell. On decomposing a hypergraph into k connected sub-hypergraphs. *Discrete Applied Mathematics*, 131(2):373–383, 2003.

[94] Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *J. ACM*, 48(6):1184–1206 (electronic), 2001.

[95] Harvey Friedman, Neil Robertson, and Paul Seymour. The metamathematics of the graph minor theorem. In *Logic and combinatorics (Arcata, Calif., 1985)*, volume 65 of *Contemp. Math.*, pages 229–261. Amer. Math. Soc., Providence, RI, 1987.

[96] Robert Ganian, Petr Hlinený, Joachim Kneis, Alexander Langer, Jan Obdrzálek, and Peter Rossmanith. On digraph width measures in parameterized algorithmics. In *4th International Workshop on Parameterized and Exact Computation (IWPEC 2009)*, pages 185–197, 2009.

[97] Archontia C. Giannopoulou, Paul Hunter, and Dimitrios M. Thilikos. Lifo-search: A min-max theorem and a searching game for cycle-rank and tree-depth. *Discrete Applied Mathematics*, 160(15):2089–2097, 2012.

[98] Archontia C. Giannopoulou, Marcin Kaminski, and Dimitrios M. Thilikos. Excluding graphs as immersions in surface embedded graphs. unpublished manuscript.

[99] Archontia C. Giannopoulou, Marcin Kaminski, and Dimitrios M. Thilikos. Forbidding kuratowski graphs as immersions. *CoRR*, abs/1207.5329, 2012.

[100] Archontia C. Giannopoulou, Sudeshna Kolay, and Saket Saurabh. New lower bound on max cut of hypergraphs with an application to r -set splitting. In David Fernández-Baca, editor, *LATIN*, volume 7256 of *Lecture Notes in Computer Science*, pages 408–419. Springer, 2012.

[101] Archontia C. Giannopoulou, Iosif Salem, and Dimitris Zoros. Effective computation of immersion obstructions for unions of graph

classes. In Fedor V. Fomin and Petteri Kaski, editors, *SWAT*, volume 7357 of *Lecture Notes in Computer Science*, pages 165–176. Springer, 2012.

[102] Archontia C. Giannopoulou, Iosif Salem, and Dimitris Zoros. Effective computation of immersion obstructions for unions of graph classes. *CoRR*, abs/1207.5636, 2012.

[103] Archontia C. Giannopoulou and Dimitrios M. Thilikos. Obstructions for tree-depth. *Electronic Notes in Discrete Mathematics*, 34:249–253, 2009.

[104] Archontia C. Giannopoulou and Dimitrios M. Thilikos. A min-max theorem for lifo-search. *Electronic Notes in Discrete Mathematics*, 38:395–400, 2011.

[105] Archontia C. Giannopoulou and Dimitrios M. Thilikos. Optimizing the graph minors weak structure theorem. *CoRR*, abs/1102.5762, 2011.

[106] Petr A. Golovach, M. Kaminski, D. Paulusma, and D. M. Thilikos. Induced packing of odd cycles in a planar graph. In *Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC 2009)*, volume 5878 of *Lecture Notes in Comput. Sci.*, pages 514–523. Springer, Berlin, 2009.

[107] Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory Comput. Syst.*, 38(4):373–392, 2005.

[108] Jens Gramm, Rolf Niedermeier, and Peter Rossmanith. Fixed-

parameter algorithms for closest string and related problems. *Algorithmica*, 37(1):25–42, 2003.

[109] Martin Grohe, Ken ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *STOC*, pages 479–488, 2011.

[110] Hermann Gruber. Digraph Complexity Measures and Applications in Formal Language Theory. In *4th Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2008)*, pages 60–67, 2008.

[111] Qian-Ping Gu and Hisao Tamaki. Improved bounds on the planar branchwidth with respect to the largest grid minor size. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *ISAAC (2)*, volume 6507 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2010.

[112] Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. A probabilistic approach to problems parameterized above or below tight bounds. *J. Comput. Syst. Sci.*, 77(2):422–429, 2011.

[113] Gregory Gutin, Leo van Iersel, Matthias Mnich, and Anders Yeo. All ternary permutation constraint satisfaction problems parameterized above average have kernels with quadratic numbers of variables. In *ESA (1)*, volume 6346 of *Lecture Notes in Computer Science*, pages 326–337, 2010.

[114] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, 10:26–30, 1935.

[115] Pinar Heggernes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. In Olaf Owe, Martin Steffen, and Jan Arne Telle, editors, *FCT*, volume 6914 of *Lecture Notes in Computer Science*, pages 240–251. Springer, 2011.

[116] Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. In Supratik Chakraborty and Amit Kumar, editors, *FSTTCS*, volume 13 of *LIPIcs*, pages 217–228. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[117] Ken ichi Kawarabayashi and Yusuke Kobayashi. The edge disjoint paths problem in eulerian graphs and 4-edge-connected graphs. In Charikar [31], pages 345–353.

[118] Ken ichi Kawarabayashi and Yusuke Kobayashi. An improved algorithm for the half-disjoint paths problem. *SIAM J. Discrete Math.*, 25(3):1322–1330, 2011.

[119] Ken ichi Kawarabayashi and Yusuke Kobayashi. List-coloring graphs without subdivisions and without immersions. In Rabani [180], pages 1425–1435.

[120] Ken ichi Kawarabayashi, Stephan Kreutzer, and Bojan Mohar. Linkless and flat embeddings in 3-space and the unknot problem. In Jack Snoeyink, Mark de Berg, Joseph S. B. Mitchell, Günter Rote, and Monique Teillaud, editors, *Symposium on Computational Geometry*, pages 97–106. ACM, 2010.

[121] Ken ichi Kawarabayashi, Zhentao Li, and Bruce A. Reed. Recognizing a totally odd k_4-subdivision, parity 2-disjoint rooted paths and

a parity cycle through specified elements. In Charikar [31], pages 318–328.

[122] Ken ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *FOCS*, pages 771–780. IEEE Computer Society, 2008.

[123] Ken ichi Kawarabayashi, Bruce A. Reed, and Paul Wollan. The graph minor algorithm with parity conditions. In Ostrovsky [178], pages 27–36.

[124] Ken ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In Leonard J. Schulman, editor, *STOC*, pages 687–694. ACM, 2010.

[125] Ken ichi Kawarabayashi and Paul Wollan. A simpler algorithm and shorter proof for the graph minor decomposition. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 451–458. ACM, 2011.

[126] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[127] Takehiro Ito, Marcin Kaminski, Daniël Paulusma, and Dimitrios M. Thilikos. Parameterizing cut sets in a graph by the number of their components. *Theor. Comput. Sci.*, 412(45):6340–6350, 2011.

[128] Weijia Jia, Chuanlin Zhang, and Jianer Chen. An efficient parameterized algorithm for m-set packing. *J. Algorithms*, 50(1):106–117, 2004.

[129] Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed tree-width. *Journal of Combinatorial Theory (Series B)*, 82(1):138–154, 2001.

[130] Marcin Kaminski and Naomi Nishimura. Finding an induced path of given parity in planar graphs in polynomial time. In Rabani [180], pages 656–670.

[131] Marcin Kaminski and Dimitrios M. Thilikos. Contraction checking in graphs on surfaces. In Christoph Dürr and Thomas Wilke, editors, *STACS*, volume 14 of *LIPIcs*, pages 182–193. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[132] Meir Katchalski, William McCuaig, and Suzanne Seager. Ordered colourings. *Discrete Math.*, 142(1-3):141–154, 1995.

[133] Ken-ichi Kawarabayashi and Yusuke Kobayashi. The induced disjoint path problem. In *Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization (IPCO 2008)*, volume 5035 of *lncs*, pages 47–61. Springer, Berlin, 2008.

[134] Ken-ichi Kawarabayashi and Bruce Reed. Odd cycle packing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 695–704, New York, NY, USA, 2010. ACM.

[135] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *CoRR*, abs/1207.0835, 2012.

[136] Eun Jung Kim and Ryan Williams. Improved parameterized algorithms for above average constraint satisfaction. In Dániel Marx and

Peter Rossmanith, editors, *IPEC*, volume 7112 of *Lecture Notes in Computer Science*, pages 118–131. Springer, 2011.

[137] Lefteris Kirousis and Christos Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(3):205–218, 1986.

[138] Yusuke Kobayashi and Ken-ichi Kawarabayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009)*, pages 1146–1155. ACM-SIAM, 2009.

[139] Athanassios Koutsonas, Dimitrios M. Thilikos, and Koichi Yamazaki. Outerplanar obstructions for matroid pathwidth. *Electronic Notes in Discrete Mathematics*, 38:541–546, 2011.

[140] Stephan Kreutzer and Sebastian Ordyniak. Digraph decompositions and monotonicity in digraph searching. In *34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008)*, pages 336–347, 2008.

[141] J. B. Kruskal. Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture. *Trans. Amer. Math. Soc.*, 95:210–225, 1960.

[142] Kazimierz Kuratowski. Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.

[143] J. Lagergren. Upper bounds on the size of obstructions and intertwines. *Journal of Combinatorial Theory. Series B*, 73:7–40, 1998.

[144] Jens Lagergren. The size of an interwine. In Serge Abiteboul and Eli Shamir, editors, *ICALP*, volume 820 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 1994.

[145] Andrea S. LaPaugh. Recontamination does not help to search a graph. *Journal of the ACM*, 40(2):224–245, 1993.

[146] F. Lescure and H. Meyniel. On a problem upon configurations contained in graphs with given chromatic number. In Lars Dovling Andersen, Ivan Tafteberg Jakobsen, Carsten Thomassen, Bjarne Toft, and Preben Dahl Vestergaard, editors, *Graph Theory in Memory of G.A. Dirac*, volume 41 of *Annals of Discrete Mathematics*, pages 325 – 331. Elsevier, 1988.

[147] Joseph W. H. Liu. The role of elimination trees in sparse factorization. *SIAM J. Matrix Anal. Appl.*, 11(1):134–172, 1990.

[148] Daniel Lokshtanov. *New Methods in Parameterized Algorithms and Complexity*. PhD thesis, Universitetet i Bergen, 2009.

[149] Daniel Lokshtanov and Saket Saurabh. Even faster algorithm for set splitting! In *IWPEC*, pages 288–299, 2009.

[150] Daniel Lokshtanov and Christian Sloper. Fixed parameter set splitting, linear kernel and improved running time. In *ACiD*, volume 4 of *Texts in Algorithmics*, pages 105–113, 2005.

[151] Meena Mahajan and Venkatesh Raman. Parameterizing above guaranteed values: Maxsat and maxcut. *J. Algorithms*, 31(2):335–354, 1999.

[152] Meena Mahajan, Venkatesh Raman, and Somnath Sikdar. Parameterizing above or below guaranteed values. *J. Comput. Syst. Sci.*, 75(2):137–153, 2009.

[153] Yury Makarychev. A short proof of Kuratowski's graph planarity criterion. *J. Graph Theory*, 25(2):129–131, 1997.

[154] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

[155] William McCuaig. Intercyclic digraphs. In Neil Robertson and Paul D. Seymour, editors, *Graph Structure Theory*, volume 147 of *Contemporary Mathematics*, pages 203–246. American Mathematical Society, 1991.

[156] Elliott Mendelson. *Introduction to mathematical logic (3. ed.)*. Chapman and Hall, 1987.

[157] Bojan Mohar and Carsten Thomassen. *Graphs on surfaces*. Johns Hopkins University Press, Baltimore, MD, 2001.

[158] C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proc. Cambridge Philos. Soc.*, 59:833–835, 1963.

[159] Jesper Nederlof and Johan M. M. van Rooij. Inclusion/exclusion branching for partial dominating set and set splitting. In *IPEC*, volume 6478 of *Lecture Notes in Computer Science*, pages 204–215, 2010.

[160] Jaroslav Nesetril and Patrice Ossona de Mendez. The grad of a graph and classes with bounded expansion. *Electronic Notes in Discrete Mathematics*, 22:101–106, 2005.

[161] Jaroslav Nesetril and Patrice Ossona de Mendez. First order properties on nowhere dense structures. *J. Symb. Log.*, 75(3):868–887, 2010.

[162] Jaroslav Nesetril and Patrice Ossona de Mendez. How many f's are there in g? *Eur. J. Comb.*, 32(7):1126–1141, 2011.

[163] Jaroslav Nesetril and Patrice Ossona de Mendez. On nowhere dense graphs. *Eur. J. Comb.*, 32(4):600–617, 2011.

[164] Jaroslav Nesetril, Patrice Ossona de Mendez, and David R. Wood. Characterisations and examples of graph classes with bounded expansion. *Eur. J. Comb.*, 33(3):350–373, 2012.

[165] Jaroslav Nešetřil and Patrice Ossona de Mendez. Linear time low tree-width partitions and algorithmic consequences. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 391–400. ACM, New York, 2006.

[166] Jaroslav Nešetřil and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European J. Combin.*, 27(6):1022–1041, 2006.

[167] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion. I. Decompositions. *European J. Combin.*, 29(3):760–776, 2008.

[168] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion. II. Algorithmic aspects. *European J. Combin.*, 29(3):777–791, 2008.

[169] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion. III. Restricted graph homomorphism dualities. *European J. Combin.*, 29(4):1012–1024, 2008.

[170] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms.* Algorithms and Combinatorics. Springer, 2012.

[171] Rolf Niedermeier. Invitation to fixed-parameter algorithms, 2002.

[172] Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.

[173] Rolf Niedermeier and Peter Rossmanith. Upper bounds for vertex cover further improved. In Christoph Meinel and Sophie Tison, editors, *STACS*, volume 1563 of *Lecture Notes in Computer Science*, pages 561–570. Springer, 1999.

[174] Rolf Niedermeier and Peter Rossmanith. An efficient fixed-parameter algorithm for 3-hitting set. *J. Discrete Algorithms*, 1(1):89–102, 2003.

[175] Rolf Niedermeier and Peter Rossmanith. On efficient fixed-parameter algorithms for weighted vertex cover. *J. Algorithms*, 47(2):63–77, 2003.

[176] Jan Obdržálek. DAG-width: Connectivity measure for directed graphs. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 814–821, 2006.

[177] Ryan O'Donnell. Some topics in analysis of boolean functions. In *STOC*, pages 569–578, 2008.

[178] Rafail Ostrovsky, editor. *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*. IEEE, 2011.

[179] T. D. Parsons. The search number of a connected graph. In *Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory, and Computing (Florida Atlantic Univ., Boca Ra-*

ton, Fla., 1978), Congress. Numer., XXI, pages 549–554, Winnipeg, Man., 1978. Utilitas Math.

[180] Yuval Rabani, editor. *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*. SIAM, 2012.

[181] Siddharthan Ramachandramurthi. The structure and number of obstructions to treewidth. *SIAM J. Discrete Math.*, 10(1):146–157, 1997.

[182] Bruce A. Reed, Neil Robertson, Paul D. Seymour, and Robin Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.

[183] Bruce A. Reed and F. Bruce Shepherd. The gallai-younger conjecture for planar graphs. *Combinatorica*, 16(4):555–566, 1996.

[184] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.

[185] Neil Robertson and P. D. Seymour. Graph minors. XXII. irrelevant vertices in linkage problems. To appear.

[186] Neil Robertson and P. D. Seymour. Graph minors—a survey. In *Surveys in combinatorics 1985 (Glasgow, 1985)*, volume 103 of *London Math. Soc. Lecture Note Ser.*, pages 153–171. Cambridge Univ. Press, Cambridge, 1985.

[187] Neil Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[188] Neil Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *J. Comb. Theory Series B*, 41:92–114, 1986.

[189] Neil Robertson and P. D. Seymour. Graph minors. VII. Disjoint paths on a surface. *J. Combin. Theory Ser. B*, 45(2):212–254, 1988.

[190] Neil Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.

[191] Neil Robertson and P. D. Seymour. Graph minors. XI. Circuits on a surface. *J. Combin. Theory Ser. B*, 60(1):72–106, 1994.

[192] Neil Robertson and P. D. Seymour. Graph minors. XII. Distance on a surface. *J. Combin. Theory Ser. B*, 64(2):240–272, 1995.

[193] Neil Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory. Series B*, 63(1):65–110, 1995.

[194] Neil Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Combin. Theory Ser. B*, 92(2):325–357, 2004.

[195] Neil Robertson, P. D. Seymour, and Robin Thomas. Linkless embeddings of graphs in 3-space. *Bulletin of the Amer. Math. Soc.*, 28(1):84–89, 1993.

[196] Neil Robertson and Paul Seymour. Excluding a graph with one crossing. In *Graph structure theory (Seattle, WA, 1991)*, pages 669–675. Amer. Math. Soc., Providence, RI, 1993.

[197] Neil Robertson, Paul Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Combin. Theory Ser. B*, 62(2):323–348, 1994.

[198] Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *J. Comb. Theory, Ser. B*, 35(1):39–61, 1983.

[199] Neil Robertson and Paul D. Seymour. Graph minors. iii. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

[200] Neil Robertson and Paul D. Seymour. Graph minors. VI. Disjoint paths across a disc. *J. Comb. Theory, Ser. B*, 41(1):115–138, 1986.

[201] Neil Robertson and Paul D. Seymour. Graph minors. IV. Tree-width and well-quasi-ordering. *J. Comb. Theory, Ser. B*, 48(2):227–254, 1990.

[202] Neil Robertson and Paul D. Seymour. Graph minors. IX. disjoint crossed paths. *J. Comb. Theory, Ser. B*, 49(1):40–77, 1990.

[203] Neil Robertson and Paul D. Seymour. Graph minors. VIII. A kuratowski theorem for general surfaces. *J. Comb. Theory, Ser. B*, 48(2):255–288, 1990.

[204] Neil Robertson and Paul D. Seymour. Graph miners .XIV. Extending an Embedding. *J. Comb. Theory, Ser. B*, 65(1):23–50, 1995.

[205] Neil Robertson and Paul D. Seymour. Graph minors: XV. Giant Steps. *J. Comb. Theory, Ser. B*, 68(1):112–148, 1996.

[206] Neil Robertson and Paul D. Seymour. Graph minors: XVII. Taming a vortex. *J. Comb. Theory, Ser. B*, 77(1):162–210, 1999.

[207] Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *J. Comb. Theory, Ser. B*, 89(1):43–76, 2003.

[208] Neil Robertson and Paul D. Seymour. Graph minors. XVIII. Tree-decompositions and well-quasi-ordering. *J. Comb. Theory, Ser. B*, 89(1):77–108, 2003.

[209] Neil Robertson and Paul D. Seymour. Graph minors. XIX. Well-quasi-ordering on a surface. *J. Comb. Theory, Ser. B*, 90(2):325–385, 2004.

[210] Neil Robertson and Paul D. Seymour. Graph minors. XXI. Graphs with unique linkages. *J. Comb. Theory, Ser. B*, 99(3):583–616, 2009.

[211] Neil Robertson and Paul D. Seymour. Graph minors XXIII. Nash-Williams' immersion conjecture. *J. Comb. Theory, Ser. B*, 100(2):181–205, 2010.

[212] Juanjo Rué, Ignasi Sau, and Dimitrios M. Thilikos. Dynamic programming for h-minor-free graphs. In Joachim Gudmundsson, Julián Mestre, and Taso Viglas, editors, *COCOON*, volume 7434 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2012.

[213] Juanjo Rué, Konstantinos S. Stavropoulos, and Dimitrios M. Thilikos. Outerplanar obstructions for a feedback vertex set. *Eur. J. Comb.*, 33(5):948–968, 2012.

[214] Mohammad Ali Safari. D-width: A more natural measure for directed tree width. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, pages 745–756, 2005.

[215] Jacob Scott, Trey Ideker, Richard M. Karp, and Roded Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. In Satoru Miyano, Jill P. Mesirov, Simon Kasif, Sorin Istrail, Pavel A. Pevzner, and Michael S. Waterman, editors, *RECOMB*, volume 3500 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2005.

[216] Paul Seymour and Paul Wollan. The structure of graphs not admitting a fixed immersion. unpublished manuscript.

[217] Paul D. Seymour and Robin Thomas. Graph searching, and a min-max theorem for tree-width. *Journal of Combinatorial Theory (Series B)*, 58:22–33, 1993.

[218] John Joseph Sylvester. Chemistry and algebra. *Nature*, 17:284, 1878.

[219] Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Minimal acyclic forbidden minors for the family of graphs with bounded path-width. *Discrete Mathematics*, 127(1/3):293–304, 1994.

[220] Dimitrios M. Thilikos. Algorithms and obstructions for linear-width and related search parameters. *Discrete Applied Mathematics*, 105:239–271, 2000.

[221] Klaus Wagner. Über eine Eigenschaft der ebenen Komplexe. *Math. Ann.*, 114(1):570–590, 1937.

[222] D. H. Younger. Graphs with interlinked directed circuits. In *Proceedings of the Midwest Symposium on Circuit Theory*, pages XVI 2.1 – XVI 2.7. Institute of Electrical and Electronics Engineers, 1973.

List of Notations

Correspondence of Terms

annulus δακτύλιος

  railed με ράγες

antichain αντιαλυσίδα

automorphism αυτομορφισμός

  λ-preserving λ-διατηρητικός

  involutive ανελικτικός

base βάση

bikernel διπυρήνας

bound φράγμα

  super-exponential υπερ-εκθετικό

  upper άνω

bramble βάτος

branch decomposition κλαδοαποσύνθεση

  width πλάτος

branch-width κλαδοπλάτος

brute force ωμή βία

chord χορδή

clique κλίκα

color χρώμα

coloring χρωματισμός

  graph γραφήματος

    proper έγκυρος

  ordered διατεταγμένος

component συνιστώσα

  connected συνεκτική

conjuctive normal form κανονική συζευκτική μορφή

contraction σύνθλιψη

  edge ακμής

  smooth στρωτή

cross-cap καπάκι διασταυρώσεων

cut τομή

cut-set διαχωριστής υπεργραφήματος

  strong ισχυρός

cycle κύκλος

  facial οψιακός

  length μήκος

cycle rank τάξη κύκλου

cylinder κύλινδρος

DAG-depth βάθος ΔΑΓ

DAG-width πλάτος ΔΑΓ

digraph διγράφημα

  strongly connected ισχυρά συνεκτικό

disjoint paths ξένα μονοπάτια
dissolution διάλυση
drawing σχεδιασμός
  orthogonal ορθογώνιος
    box κουτί
    joining vertex συνδετική κορυφή

edge ακμή
  incident προσπίπτουσα
  parallel παράλληλη
edge lift ανύψωση ακμών
edge-cut ακμοδιαχωριστής
  internal εσωτερικός
  minimal ελαχιστικός
edge-linkage ακμοδέσμωση
  r**-approximate** r-προσεγγιστική
  component συνιστώσα
  links συνδέει
  order τάξη
edge-sum ακμοάθροισμα
embedding εμβάπτιση
  2-cell 2-κελιών

family οικογένεια
  cycle κύκλων
    nested εμφωλευμένων
  paths μονοπατιών
    confluent σύρροη
forest δάσος

fugitive φυγάς

  invisible αόρατος

  omniscient σοφός

  visible ορατός

graph γράφημα

  apex απόγειο

  2-asymmetric 2-ασύμμετρο

  asymmetric ασύμμετρο

  bipartite διμερές

    complete πλήρες

  chordal χορδικό

  complete πλήρες

  connected συνεκτικό

  directed διατεταγμένο

    strongly connected ισχυρά συνεκτικό

  edge-connected ακμοσυνεκτικό

  Σ-embeddable Σ-εμβαπτίσιμο

  Σ_0-embedded Σ_0-εμβαπτισμένο

  empty κενό

  incidence πρόσπτωσης

  line γραμμικό

  H-minor-free H-ελεύθερο-ελάσσονος

  h-nearly embeddable h-σχεδόν εμβαπτίσιμο

  planar επίπεδο

    face όψη

  plane ενεπίπεδο

  primal πρωτογενές

  subcubic υποκυβικό

triangulated τριγωνοποιημένο

triangulation τριγωνοποίηση

graph class κλάση γραφημάτων

bounded expansion φραγμένης επέκτασης

covering κάλυμμα

H**-minor-free** H-ελεύθερη-ελάσσονος

MSO-definable ΜΔΛ-ορίσιμη

layer-wise κάτα στρώσεις

packing συσκευασία

width πλάτος

graph connectivity συνεκτικότητα γραφήματος

graph edge-connectivity ακμοσυνεκτικότητα γραφήματος

graph searching ανίχνευση γραφημάτων

graph separator διαχωριστής γραφήματος

graphs γραφήματα

disjoint ξένα

edge- ως προς ακμές

vertex- ως προς κορυφές

disjoint union ξένη ένωση

hom-equivalent ομο-ισοδύναμα

homomorphic ομομορφικά

intersection τομή

isomorphic ισομορφικά

union ένωση

grid σχάρα

corner γωνία

loaded φορτωμένη

vertex κορυφή

external εξωτερική
internal εσωτερική

haven άσυλο
LIFO- ΤΕΠΕ-
homomorphism ομομορφισμός
hyperedge υπερακμή
split χωρισμένη
hypergraph υπεργράφημα
connected συνεκτικό
partition ως προς διαμερίσεις
hypertree υπερδέντρο
hypothesis υπόθεση
exponential time εκθετικού χρόνου

isomorphism ισομορφισμός

kernel πυρήνας
linear γραμμικός
polynomial πολυωνυμικός
kernelization πυρηνοποίηση

linear-width γραμμοπλάτος
linkage δέσμωση
r**-approximate** r-προσεγγιστική
component συνιστώσα
links συνδέει
order τάξη
unique μοναδική
vital ζωτική

logic λογική

arity πλειομέλεια

first-order πρωτοβάθμια

syntax σύνταξη

interpretation ερμηνεία

second-order δευτεροβάθμια

atomic formula ατομικός τύπος

monadic μοναδική

syntax σύνταξη

structure δομή

universe σύμπαν

loop θηλιά

matching ταίριασμα

saturated κορεσμένο

max cut μέγιστη τομη

max sat μέγιστη ΙΛΤ

minor containment περιεκτικότητα ελάσσονος

multigraph πολυγράφημα

multiset πολυσύνολο

NAE-SAT ΟΟΙ-ΙΛΤ

neighborhood γειτονιά

closed κλειστή

open ανοικτή

number αριθμός

chromatic χρωματικός

list λίστας

covering καλύμματος

cycle dominating κυριαρχίας κύκλων
r**-dominating** r-κυριαρχίας
packing συσκευασίας
scattered cycle σκεδασμένων κύκλων
r**-scattering** r-σκέδασης
search ανίχνευσης

obstruction παρεμπόδιση
orbit τροχιά
ordering διάταξη
partial μερική
well-quasi-ordering καλή

parameterization παραμετροποίηση
natural τυπική
part of bipartition μέρος διαμέρισης
path μονοπάτι
endpoints άκρα
path-width πλάτος μονοπατιού
directed διατεταγμένο
proper γνήσιο
paths μονοπάτια
confluent σύρροα
well-arranged καλώς-κατεταγμένα
problem πρόβλημα
fixed parameter tractable παραμετρικά βατό
parameterized παραμετροποιημένο
product γινόμενο
cartesian καρτεσιανό

- **lexicographic** λεξικογραφικό

property ιδιότητα

- **fixed point** σταθερού σημείου

pyramid πυραμίδα

rails ράγα

reduction αναγωγή

reduction rule κανόνας αναγωγής

relation σχέση

- **contraction** σύνθλιψη
  - **model** μοντέλο
- **immersion** εμβύθιση
  - **strong** ισχυρή
- **minor** ελάσσον
  - **minimal model** ελαχιστικό μοντέλο
- **subdivision** υποδιαίρεση
- **topological minor** τοπολογικό ελάσσον
  - **distance** απόστασης

3-SAT 3-ΙΛΤ

searcher ανιχνευτής

separation triple διαχωριστική τριάδα

sequence ακολουθία

- **augmenting** επαυξητική

set σύνολο

- **dominating** κυριαρχίας
- **edge** ακμών
- **obstruction** παρεμπόδισης
  - **immersion** εμβυθίσεων

minor ελασσόνων
siblings αδερφών κορυφών
vertex κορυφών
feedback ανάδρασης
set splitting διαχωρισμός συνόλων
above average άνω μέσου όρου
shelter καταφύγιο
directed διατεταγμένο
thickness πάχος
thickness πάχος
spanner παράγοντας
multiplicative πολλαπλασιαστικός
stretch factor συντελεστής έκτασης
F**-split** F-σχίσμα
star αστέρι
strategy στρατηγική
monotone μονότονη
non-monotone μη-μονότονη
searcher stationary σταθμευμένων ανιχνευτών
subgraph υπογράφημα
induced εναγόμενο
spanning παραγόμενο
surface επιφάνεια
compact συμπαγής
Eüler genus γένος Eüler
manifold πολλαπλότητα
boundary όριο
orientable προσανατολισμένη

Eüler genus γένος Eüler

technique τεχνική

irrelevant vertex μη-σχετικής

theorem θεώρημα

excluded grid αποκλεισμένης σχάρας

four color τεσσάρων χρωμάτων

graph minor ελασσόνων γραφημάτων

meta-algorithmic μετα-αλγοριθμικό

structure δομικό

strong ισχυρό

weak ασθενές

theory θεωρία

bidimensionality δισδιαστατότητας

graph minor ελασσόνων γραφημάτων

parameterized complexity παραμετρικής πολυπλοκότητας

tiling πλακόστρωση

monohedral μονοεδρική

regular κανονική

tour περιοδεία

tree δέντρο

detachment αποκόλλησης

elimination διαγραφής

minimum height ελάχιστου ύψους

leaves φύλλα

ternary τριαδικό

tree decomposition δεντροαποσύνθεση

bag τσάντα

closure κλειστότητα

node κορυφή

base βάσης

forget λήθης

introduction εισαγωγής

join σύνδεσης

small μικρή

torso κορμός

width πλάτος

tree-dec expansion δεντρό-επέκταση

tree-depth βάθος δέντρου

tree-width δεντροπλάτος

directed διατεταγμένο

vertex κορυφή

degree βαθμός

overlapping επικαλύπτουσα

vertex cover κάλυμμα κορυφών

vertex ranking διαβάθμιση κορυφών

vertices κορυφές

adjacent γειτονικές

apices απόγειες

distance απόσταση

vortex στρόβιλος

walk περίπατος

wall τοίχος

brick τούβλο

neighboring γειτονικά

compass περιφέρεια

corner γωνία

  antidiametrical pairs αντιδιαμετρικά ζεύγη

flat ισόπεδος

height ύψος

layer στρώμα

paths μονοπάτια

  eastern ανατολικότερο

  horizontal οριζόντια

  northern βορειότερο

  southern νοτιότερο

  vertical κάθετα

  western δυτικότερο

perimeter περίμετρος

rural division περιφερειακή διαίρεση

  flap πτερύγιο

  internal flap εσωτερικό πτερύγιο

subdivided υποδιαιρεμένος

subdivision υποδιαίρεση

tight σφικτός

vertex κορυφή

  important σημαντική

  original πρωταρχική

  subdivision υποδιαίρεσης

Index