UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

DOKTORSKA DISERTACIJA
(DOCTORAL THESIS)

STRUKTURA IN ALGORITMI V RAZREDIH GRAFOV: NOVI REZULTATI O MINIMALNIH SEPARATORJIH IN NEODVISNIH MNOŽICAH

(STRUCTURE AND ALGORITHMS FOR GRAPH CLASSES: NEW RESULTS ON MINIMAL SEPARATORS AND INDEPENDENT SETS)

NEVENA PIVAČ

KOPER, 2024

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

DOKTORSKA DISERTACIJA
(DOCTORAL THESIS)

STRUKTURA IN ALGORITMI V RAZREDIH GRAFOV: NOVI REZULTATI O MINIMALNIH SEPARATORJIH IN NEODVISNIH MNOŽICAH

(STRUCTURE AND ALGORITHMS FOR GRAPH CLASSES: NEW RESULTS ON MINIMAL SEPARATORS AND INDEPENDENT SETS)

NEVENA PIVAČ

KOPER, 2024 MENTOR: PROF. DR. MARTIN MILANIČ

Za svu snagu kada ustadoh pa padoh,
za noći kad bez odgovora dozivaste mama.
I za te dane kad sebe vam kradoh
svako slovo ovdje ja poklanjam vama.

Dunji i Vasiliju

Acknowledgements

I would like to express my gratitude to my mentor, Prof. Martin Milanič for his guidance, motivation and support. He has been imparting knowledge to me for over a decade and has provided me with many opportunities to broaden my professional network. I am grateful to people from UP IAM and UP FAMNIT for giving me the opportunity to learn and for understanding all the needs that arose during this journey. Special thanks go to Nina, Matjaž, Vito, Klavdija, Dragan, Ademir, and all supporting staff, as well as to the people I have worked with, in particular: Ekki's group in Cottbus, Ulrich's group in Graz, and Irena Penev.

Words cannot express my gratitude to my family for their support and love throughout my life. Firstly, thanks go to my husband Darko that has pushed the boundaries of my faith by showing me that everything is possible if the desire is strong enough. Thank you for being my support, for making me feel alive, for loving me, and for believing in me when I didn't even believe in myself.

I am grateful to my little sisters for their love, inspiring conversations and unconditional support. Furthermore, I am grateful to Lazo for his immense respect and help, as well as to my father and my mother-in-law for being always there when needed.

And the ultimate meaning of all this is given by two little hearts. Dunja and Vasilije, thank you for giving meaning to my every breath and for the incredible moments we spend together. Everything here is for you and because of you. Be proud of me, of us, and let this motivate you when success seems too far away. Our small family is my source of motivation.

Without my friends everything would be much more difficult. One of the remarkable individuals in my life is my friend Šejla and I am grateful to her for all the wonderful memories from student days, as well as for the unconditional support during the hardest moments for me and my family.

For the crucial support in the most difficult moments, I would like to thank Dr. Lilijana Kornhauser Cerar, Dr. Matej Furlan, Nurse Jelka, Manca Oblak and the other medical staff in University Medical Centre Ljubljana.

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

STRUCTURE AND ALGORITHMS FOR GRAPH CLASSES: NEW RESULTS ON MINIMAL SEPARATORS AND INDEPENDENT SETS

This doctoral thesis explores diverse classes of graphs across two principal parts. The first part is dedicated to the study of minimal separators, while in the subsequent part we consider some old and new algorithmic problems related to independent sets.

A *minimal separator* of a graph G is a set $S \subseteq V(G)$ such that there exist vertices $a, b \in V(G) \setminus S$ with the property that S separates a from b in G, but no proper subset of S does. A graph class is said to be *tame* if graphs in the class have a polynomially bounded number of minimal separators. Tame graph classes have good algorithmic properties, which follow, for example, from an algorithmic metatheorem of Fomin, Todinca, and Villanger from 2015. We show that a hereditary graph class G is tame if and only if the subclass consisting of graphs in G without clique cutsets is tame. This result and Ramsey's theorem lead to several types of sufficient conditions for a graph class to be tame. We apply these results, combined with the structure of graphs with exponentially many minimal separators, to develop dichotomy theorems separating tame from non-tame graph classes within the families of graph classes defined by sets of forbidden induced subgraphs with at most four vertices. Building on recent works of Gartland and Lokshtanov [SODA 2023], and of Gajarský et al. [arXiv, 2022], we characterize tame graph classes defined by a single forbidden induced minor or induced topological minor. We provide polynomial-time recognition algorithms for the maximal tame graph classes obtained in the above characterizations by a single forbidden induced minor or induced topological minor.

Among the tame classes of graphs of special interest are classes of graphs having a linear number of minimal separators, which makes the algorithms based on minimal separators particularly efficient. We consider a number of well-known and interrelated classes of graphs having the number of minimal separators bounded by the number of vertices: threshold graphs, cographs, split graphs, pseudo-split graphs, trivially perfect graphs, co-trivially perfect graphs, and $2P_2$-free graphs. For each of these classes, we establish exact values for the maximum number of minimal separators in an n-vertex graph from the class.

For an integer $k \geq 0$, we say that a minimal separator is *k-simplicial* if it is a union of k cliques and denote by $\mathcal{G}_k$ the class of all graphs in which each minimal separator is k-simplicial. We show that for each $k \geq 0$, the class $\mathcal{G}_k$ is closed under induced minors and give a complete list of minimal forbidden induced minors for $\mathcal{G}_2$. A known result on chordal graphs states that every chordal graph has a simplicial vertex. We generalize this result by showing that for $k \geq 1$ every LexBFS ordering of a graph in $\mathcal{G}_k$ is a k-simplicial elimination ordering and consequentially, every nonnull graph in $\mathcal{G}_k$ has a k-simplicial vertex, that is, a vertex whose neighborhood is a union of k cliques. Further, we show that, for $k \geq 3$, it is NP-hard to recognize graphs in $\mathcal{G}_k$. The time complexity of recognizing graphs in $\mathcal{G}_2$ is unknown. We study various subclasses of $\mathcal{G}_2$ and obtain polynomial-time recognition algorithms for these restricted cases. Finally, we show a number of algorithmic results for the class $\mathcal{G}_2$ and its subclasses.

In the second part of the thesis we consider two distinct problems, related to INDEPENDENT SET, the following well-known NP-complete problem: given a graph G and an integer k, determine whether G contains an independent set of cardinality k. Firstly, we study the problem of computing the vector space consisting of all well-covered weightings of a graph G, that is, of all vertex weight functions on the graph under which all the maximal independent sets of the graph have constant weight. This set forms a vector space over the field of real numbers, called the well-covered vector space of G. The problem of computing the well-covered vector space of a given graph is co-NP-hard. In the thesis we give two general reductions for the problem, one based on anti-neighborhoods and one based on modular decomposition, combined with Gaussian elimination. Building on these results, we develop a polynomial-time algorithm for computing the well-covered vector space of a given fork-free graph, generalizing the result of Levit and Tankus that solves the problem for claw-free graphs.

The second problem studied in this part of the thesis is an allocation problem: a problem of computing an optimal allocation of items to agents in the presence of a conflict graph, respecting a certain fairness criterion. We study the fair allocation of indivisible items to several agents and introduce an incompatibility relation between pairs of items described in terms of a *conflict graph*. Every subset of items assigned to one agent has to form an independent set in this graph and every agent has its own profit valuation for every item. Aiming at a fair allocation, the goal is the maximization of the lowest total profit of items allocated to any one of the agents. We derive complexity and algorithmic results depending on the properties of the given graph. We show that the problem is strongly NP-hard for bipartite graphs and their line graphs, and solvable in pseudo-polynomial time for the classes of chordal graphs, cocomparability graphs, biconvex bipartite graphs, graphs of bounded treewidth, and graphs of bounded clique-width.

Math. Subj. Class (2010): 05C69, 05C75, 05C35, 05C83, 05C85, 90C39, 90C47, 90C27, 91B32

Key words: minimal separator, bisimplicial separator, structural characterization of families of graphs, graph class, tame graph class, fair allocation, fork-free graph, well-covered vector space, independent set, graph algorithm, induced subgraph, induced minor, induced topological minor.

Povzetek

STRUKTURA IN ALGORITMI V RAZREDIH GRAFOV: NOVI REZULTATI O MINIMALNIH SEPARATORJIH IN NEODVISNIH MNOŽICAH

Doktorska disertacija preučuje različne razrede grafov in je strukturirana preko dveh glavnih delov. Prvi del je posvečen študiju minimalnih separatorjev, medtem ko v drugem delu študiramo nekatere stare in nove algoritmične probleme, povezane z neodvisnimi množicami.

Minimalnen separator grafa G je taka množica $S \subseteq V(G)$, za katero obstajata točki $a, b \in V(G) \setminus S$, ki imata lastnost, da S loči a od b v G, vendar to ne velja za nobeno pravo podmnožico množice S. Razredu grafov pravimo, da je *krotek*, če imajo grafi v razredu polinomsko omejeno število minimalnih separatorjev. Krotki razredi grafov imajo dobre algoritmične lastnosti, ki izhajajo, na primer, iz algoritmičnega metaizreka Fomina, Todince in Villangerja iz leta 2015. V disertaciji pokažemo, da je hereditaren razred grafov $\mathcal{G}$ krotek natanko takrat, ko je podrazred, sestavljen iz grafov v $\mathcal{G}$ brez prereznih klik, krotek. Ta rezultat in Ramseyjev izrek vodita do več vrst zadostnih pogojev za to, da je razred grafov krotek. Omenjeni rezultati, skupaj s strukturo grafov z eksponentno mnogo minimalnimi separatorji, vodijo do dihotomij, ki loči krotke razrede grafov od nekrotkih znotraj družin razredov grafov, definiranih s seznamom prepovedanih induciranih podgrafov z največ štirimi točkami. Na podlagi nedavnega dela Gartlanda in Lokshtanova [SODA 2023] ter Gajarskega idr. [arXiv, 2022] karakteriziramo krotke razrede grafov, definirane z enim prepovedanim induciranim minorjem oz. induciranim topološkim minorjem. Poleg tega razvijemo polinomske algoritme za prepoznavanje maksimalnih krotkih razredov grafov, pridobljenih v omenjenih karakterizacijah s posameznim prepovedanim induciranim minorjem oz. induciranim topološkim minorjem.

Med krotkimi razredi grafov so še posebej zanimivi razredi grafov, v katerih je število minimalnih separatorjev omejeno z linearno funkcijo, zaradi česar so algoritmi, ki temeljijo na minimalnih separatorjih, še posebej učinkoviti. V disertaciji študiramo več znanih in medsebojno povezanih razredov grafov, ki imajo število minimalnih separatorjev omejeno s številom točk: pragovni grafi, kografi, razcepljeni grafi, psevdorazcepljeni grafi, trivialno popolni grafi, ko-trivialno popolni grafi in $2P_2$-prosti grafi. Za vsakega od teh razredov določimo natančne vrednosti za največje število minimalnih separatorjev v n-točkovnem grafu, ki pripada razredu.

Za celo število $k \geq 0$ pravimo, da je minimalen separator *k-simplicialen*, če je unija k klik, in z $\mathcal{G}_k$ označimo razred vseh grafov, v katerih je vsak minimalen separator k-simplicialen. Pokažemo, da je za vsak $k \geq 0$ razred $\mathcal{G}_k$ zaprt za inducirane minorje in podamo popoln seznam minimalnih prepovedanih induciranih minorjev za $\mathcal{G}_2$. Znano je, da vsak tetiven graf vsebuje simplicialno točko. V doktorski disertaciji ta rezultat posplošimo tako, da pokažemo, da za $k \geq 1$ vsaka LexBFS razvrstitev točk grafa v $\mathcal{G}_k$

predstavlja k-simplicialno eliminacijsko shemo. Posledično, vsak neprazen graf v $\mathcal{G}_k$ vsebuje k-simplicialno točko. Nadalje pokažemo, da je za vsak $k \geq 3$ NP-težko prepoznati grafe v G_k. Časovna zahtevnost prepoznavanja grafov v $\mathcal{G}_2$ je odprt problem, zato študiramo različne podrazrede razreda $\mathcal{G}_2$ in razvijemo polinomske algoritme prepoznavanja grafov znotraj teh podrazredov. Na koncu pokažemo nekaj algoritmičnih rezultatov za razred $\mathcal{G}_2$, kot tudi za določene podrazrede razreda $\mathcal{G}_2$.

V drugem delu disertacije študiramo dva različna problema, povezana s problemom NEODVISNE MNOŽICE, znanim NP-polnim problemom: če imamo podan graf G in celo število k, določi, ali G vsebuje neodvisno množico moči k. Prvič, študiramo problem izračuna vseh dobrih pokritij grafa, to je, vseh utežnih funkcij na točkah grafa, glede na katere imajo vse maksimalne neodvisne množice grafa enako težo. Množica vseh dobrih pokritij grafa G tvori vektorski prostor nad poljem realnih števil, ki mu rečemo *dobro pokrit prostor* grafa G. Problem izračuna dobro pokritega vektorskega prostora danega grafa je co-NP-težek. V disertaciji podamo dve splošni prevedbi za ta problem: eno, ki temelji na nesoseščinah v grafu, in drugo, ki temelji na modularni dekompoziciji, v kombinaciji s Gaussovo eliminacijo. Na podlagi teh rezultatov razvijemo polinomski algoritem za izračun dobro pokritega vektorskega prostora danega grafa brez vilic, ki posploši rezultat Levita in Tankusa, kjer se problem reši za grafe brez krempljev.

Drugi problem, ki se mu posvečamo v tem delu disertacije, je problem poštene razdelitve: problem izračuna optimalne razdelitve nedeljivih predmetov agentom, z upoštevanjem konfliktnega grafa in določenih omejitev poštene razdelitve. S konfliktnim grafom lahko prepovemo hkratno uporabo določenih predmetov, tako da vsaka množica predmetov, ki je dodeljena določenemu agentu, predstavlja neodvisno množico konfliktnega grafa. Vsak agent ima svojo profitno funkcijo na množici vseh predmetov in naš cilj je maksimizacija najnižjega posameznega profita poljubne množice predmetov, dodeljenih kateremu koli od agentov. V disertaciji pokažemo, da je omenjeni problem krepko NP-težek za dvodelne grafe in njihove povezavne grafe, ter rešljiv v psevdopolinomskem času za razrede tetivnih grafov, neprimerljivostnih grafov, bikonveksnih dvodelnih grafov, grafov omejene drevesne širine in grafov omejene klične širine.

Math. Subj. Class (2010): 05C69, 05C75, 05C35, 05C83, 05C85, 90C39, 90C47, 90C27, 91B32

Ključne besede: minimalen separator, bisimplicilen separator, strukturna karakterizacija družin grafov, razred grafov, krotki razred grafov, poštena razvrstitev, graf brez vilic, dobro pokrit vektorski prostor, neodvisna množica, algoritmi na grafih, induciran podgraf, induciran minor, induciran topološki minor.

Chapter 1

Introduction

A graph is a simple concept in mathematics that consists of points and lines connecting them, and enables us to represent many real-world problems. Consider, for example, any social network where the accounts are the points, and the connections are the lines, or a roadmap, where the cities are the points, and the roads are the lines. Clearly, these can be represented by graphs. Formally, a *graph* G consists of a vertex set $V = V(G)$ and edge set $E = E(G)$, with edges being unordered pairs of distinct vertices. Graphs represent a mathematical structure used to model pairwise relations between objects. In particular, they can be used to model various relations and processes in physical, biological, social, and information systems. The very first paper in the history of graph theory was published in 1736, by Leonhard Euler [100], while the term "graph" was introduced by Sylvester in a paper published in 1878 [222].

Over the years, many problems were modelled by graphs, and consequentially, various graph concepts were introduced and studied. Of particular interest are subsets of vertices that possess a certain property, and their cardinalities. A *minimal separator* in a graph is an inclusion-minimal set of vertices whose removal disconnects a fixed pair of non-adjacent vertices. An *independent set* in a graph G is a set of pairwise non-adjacent vertices. This doctoral thesis is divided into two parts. The first part is dedicated to the study of minimal separators, while the second one considers some old and new algorithmic problems related to independent sets. The central theme common to both parts is the study of graph classes, that is, sets of graphs closed under isomorphism.

1.1 Minimal separators

A *minimal separator* of a graph G is a set $S \subseteq V(G)$ such that there exist vertices $a, b \in V(G) \setminus S$ with the property that S separates a from b in G, but no proper subset of S does. Note that it is possible that S is a minimal separator of a graph G, even though some $S' \subsetneqq S$ is also a separator of G. Indeed, there may be a pair a, b of non-adjacent vertices such that S is a minimal (a, b)-separator of G, as well as some other pair a', b' of non-adjacent vertices such that some $S' \subsetneqq S$ is an (a', b')-separator of G.

Minimal separators have been studied since at least the 1960s, when chordal graphs (graphs without any induced cycle of length at least four) were characterized as precisely those graphs in which all minimal separators are cliques (sets of paiwise adjacent vertices) [95]. Chordal graphs are a well-studied graph class for many reasons; they have good structural properties, which imply efficient algorithms for many problems that are NP-hard in general. For example, finding a maximum independent set can be done in polynomial time in the class of chordal graphs. This motivated the study of various graph classes in similar context—to obtain a characterization with minimal separators. For instance, Backer characterized interval, cocomparability, and AT-free graphs by the structure of minimal separators [12].

Minimal separators were subsequently studied in [24] in the context of moplexes, have played an important role in sparse matrix computations via minimal triangulations (for a survey, see [139]), and have also had numerous algorithmic applications (see, e.g., [33, 41, 226]). Many graph algorithms and characterizations are based on minimal separators (see, e.g., [12, 24, 33, 34, 35, 41, 62, 76, 95, 153, 193]) and such algorithms usually enumerate all minimal separators of the input graph at some step of execution (see, e.g. [42, 112, 185]). Thus, the number of minimal separators directly influences their running time. Fomin et al. proved in [110] that the maximum number of minimal separators over all n-vertex graphs is bounded from below by $\Omega(3^{n/3})$ and from above by $\mathcal{O}(1.708^n)$. This bound was improved independently by Fomin and Villanger [113] and by Gaspers and Mackenzie [123]. They showed that the maximum number of minimal separators in a graph is bounded from above by $\mathcal{O}(\rho^n n)$, where $\rho = (1+\sqrt{5})/2$. It turns out that for the classes of graphs with maximum number of minimal separators of n-vertex graphs bounded by a polynomial in n, all of the algorithms from [42, 112, 185] run in polynomial time, so many problems that are NP-hard for general graphs become polynomial-time solvable for classes of graphs with polynomially bounded number of minimal separators. This is the case for TREEWIDTH and MINIMUM FILL-IN [42], for MAXIMUM INDEPENDENT SET, FEEDBACK VERTEX SET, and more generally the problem of finding a maximum induced subgraph of treewidth at most a constant t [112], and for DISTANCE-d INDEPENDENT SET for even d [185].

The nice properties of classes of graphs having polynomially many minimal separators that lead to the existence of efficient algorithms for various graph problems motivate the classification of such graph classes. In this thesis, we characterize classes of graphs with polynomially bounded number of minimal separators within certain families of graph classes, and exactly determine the number of minimal separators in various graph classes. Furthermore, we generalize the concept of chordal graphs by considering graphs where every minimal separator is a union of a bounded number of cliques and study the algorithmic consequences for several classical optimization graph problems when restricted to such graph classes.

1.2 Independent sets

Given a graph G and an integer k, deciding whether G contains an independent set of cardinality k is an NP-complete problem known under the name INDEPENDENT SET [146]. If every vertex of a graph G is assigned a real number, the *weight* of a vertex, we speak about a *weighted graph*. Given a graph G and a weight function $w : V(G) \to \mathbb{R}$, the weight of any set $S \subseteq V(G)$ is defined as $w(S) = \sum_{v \in S} w(v)$. A natural generalization of INDEPENDENT SET is MAXIMUM WEIGHT INDEPENDENT SET, that is, the problem of computing an independent set of maximum weight in a given weighted graph.

An independent set in a graph G is said to be *maximal* if it is not properly contained in any larger independent set of G, and *maximum*, if there is no independent set in G having larger cardinality. In the thesis we consider two distinct problems related to Independent Set. The first one is the problem of computing the vector space consisting of all vertex weight functions under which all the maximal independent sets of the graph have constant weight. The second one is an allocation problem: a problem of computing an optimal allocation of items to agents in the presence of a conflict graph, respecting a certain fairness criterion. Note that the conflict graph here enables us to forbid the simultaneous usage of particular items, so every subset of items assigned to one agent has to form an independent set in the conflict graph.

Well-covered weightings of a graph

A graph is *well-covered* if all its maximal independent sets have the same cardinality. This concept was introduced by Plummer in 1970 [201] and naturally generalizes to the weighted case. A weighted graph G is said to be w-*well-covered* if all maximal independent sets in G are of the same weight with respect to the weight function w. This concept was introduced by Caro, Ellingham, and Ramey in 1998 [52], in the more general context of weight functions mapping the vertices of a graph to the elements of an abelian group (see also [50]).

In the thesis we study *well-covered weighting* of a graph G, that is, real-valued weight functions w on the vertices of G such that G is w-well-covered. It is known that for every graph G, the set $\mathrm{WCW}(G)$ of all well-covered weightings of G forms a vector space over the field of real numbers (see [50,54]); we refer to it as the *well-covered vector space* of G. Any system of linear equations describing the vector space $\mathrm{WCW}(G)$ will be referred to as a *well-covering system* of G. We consider the problem of determining a well-covering system of a graph G, and refer to this problem as Well-Covering System.

Since the problem of recognizing well-covered graphs is co-NP-complete (see [68, 214]), the more general Well-Covering System problem is co-NP-hard. Well-covered vector spaces of various graph classes were studied over the years (see, e.g., [51, 52,54,165]). In particular, Levit and Tankus showed that the problem can be solved in polynomial time in the class of claw-free graphs [164] (where a *claw* is a graph on four vertices, with one vertex of degree 3 and three vertices of degree 1). We continue this line of research and study the complexity of Well-Covering System in the class of fork-free graphs (where a *fork* is a claw with one edge subdivided). We give two general reductions for the problem, one based on anti-neighborhoods and one based on modular decomposition, combined with Gaussian elimination. Building on these results, we develop a polynomial-time algorithm for computing the well-covered vector space of a given fork-free graph, generalizing the result of Levit and Tankus. Our approach implies a polynomial-time recognition algorithm for the class of well-covered fork-free graphs and also generalizes some known results on cographs (see [50]).

Fair allocation with conflict graphs

Distributing resources among multiple agents effectively is a classic problem in combinatorial optimization. In this scenario, each agent assigns a value to each item, and the objective is to assign every item to exactly one agent in a way that maximizes the lowest value assigned by any agent (see, e.g., [44, 227]). Usually, such problems are equipped with some additional constraints for a feasible allocation, and there are various models of preferences expressed by the agents and different objectives arising from these. If there are no other restrictions, the proposed classical problem is called FAIR k-DIVISION OF INDIVISIBLE ITEMS: allocate items to k agents, where each agent has its own utility function over the set of items, and the goal is to maximize the utility over all agents.

It is, however, often the case that a set of feasible solutions of an allocation of resources is restricted by various constraints. In particular, any incompatibility relation between pairs of items can be modelled using a conflict graph. The vertices of a conflict graph are items and an incompatibility among items i and j is modelled by the presence of the edge ij in the conflict graph. The items are in conflict if they cannot be used simultanuosly, or simply cannot be allocated to the same agent, for any reason. Therefore, we study the fair allocation of n indivisible goods or items to a set of k agents from a graph theoretical perspective. Every agent has its own value for each item, and assuming that a conflict graph is given, we want to compute a fair allocation of items to agents such that the minimal total weight among agents is maximized. This problem will be called FAIR k-DIVISION UNDER CONFLICTS. Note that every subset of items assigned to one agent has to form an independent set in a conflict graph, and it can happen that the allocation is partial (not all items are allocated).

Recent papers from this field studying fairness issues in connection with the underlying graph structure are given in [16,43]. Conflict-free allocation of items immediately leads to partial colorings of the conflict graph (see [22,90]).

Note that for $k = 1$, the problem coincides with the MAXIMUM WEIGHT INDEPENDENT SET. In particular, since the case of unit weights and $k = 1$ generalizes the INDEPENDENT SET problem, we conclude that FAIR 1-DIVISION UNDER CONFLICTS is strongly NP-hard. With some effort, one can see that even without conflicts the FAIR k-DIVISION OF INDIVISIBLE ITEMS problem is weakly NP-hard for any constant $k \geq 2$ and strongly NP-hard for k being part of the input. This holds even for k identical profit functions. Thus, unless P = NP, pseudo-polynomial algorithms for FAIR k-DIVISION UNDER CONFLICTS can only be developed for constant k.

In the thesis we give a characterization of the computational complexity of FAIR k-DIVISION UNDER CONFLICTS for different classes of conflict graphs. We study the boundary between strongly NP-hard cases and those where a pseudo-polynomial algorithm can be derived for constant k.

1.3 Publications

Some results presented in this doctoral thesis appear as part of submitted preprint or published journal papers, prepared in collaboration with my advisor and/or with some other collaborators. During my studies, I have also co-authored papers that are not part of this doctoral thesis (marked by * below).

[18]* J. Beisegel, C. Denkert, E. Köhler, M. Krnc, N. Pivač, R. Scheffler, and M. Strehler. *On the end-vertex problem of graph searches.* Discrete Math. Theor. Comput. Sci., 21(1):20, 2019. Id/No 13.

[19]* J. Beisegel, C. Denkert, E. Köhler, M. Krnc, N. Pivač, R. Scheffler, and M. Strehler. *The recognition problem of graph search trees.* SIAM J. Discrete Math., 35(2):1418–1446, 2021

[59]* N. Chiarelli, C. Dallard, A. Darmann, S. Lendl, M. Milanič, P. Muršič, U. Pferschy, and N. Pivač. *Allocation of indivisible items with individual preference graphs.* Discret. Appl. Math., 334:45–62, 2023.

[60] N. Chiarelli, M. Krnc, M. Milanič, U. Pferschy, N. Pivač, and J. Schauer. *Fair allocation of indivisible items with conflict graphs.* Algorithmica, 85(5):1459–1489, 2023.

[180] M. Milanič, I. Penev, N. Pivač, and K. Vušković. *Bisimplicial separators.* J.Graph Theory, 2024. doi:10.1002/jgt.23098.

[181] M. Milanič and N. Pivač. *Computing well-covered vector spaces of graphs using modular decomposition.* Comput. Appl. Math., 42(8):360, 2023.

[183] M. Milanič and N. Pivač. *Polynomially bounding the number of minimal separators in graphs: reductions, sufficient conditions, and a dichotomy theorem.* Electron. J. Combin., 28(1):Paper No. 1.41, 27, 2021.

1.4 Structure of the thesis

In Chapter 2 we summarize the main definitions, terminology and preliminary results used throughout the thesis. The rest of the thesis is divided into two parts, regarding the main topic. Part I studies minimal separators and consists of four chapters. In Chapter 3 we give some motivation and a general overview of the problems studied in Part I, as well as some preliminary results. The first problem, studied in Chapter 4, deals with tame graph classes, that is, graph classes with a polynomially bounded number of minimal separators. In Chapter 5 we study the extremal number of minimal separators in graph classes where this number is bounded by the number of vertices. In Chapter 6 we study classes of graphs where each minimal separator is a union of k cliques (k-simplicial). Concluding remarks and potential directions for further research regarding the problems studied in Part I are given in Chapter 7.

Part II has similar structure as Part I: general remarks are presented in Chapter 8, while the two problems considered in this part are presented in Chapters 9 and 10. The former chapter considers the problem of determining a well-covering system of fork-free graphs, while the latter one considers the fair allocation problem with a conflict graph. Final remarks to Part II are given in Chapter 11.

Chapter 2

Preliminaries

In this chapter we provide the basic notation and definitions, recall the basic properties of some graph classes relevant to our study, and show some preliminary results that will be used throughout the thesis. Some concepts are defined again later in the thesis but are collected here for the convenience of the reader.

2.1 General preliminaries on graphs

All graphs in this paper will be finite, simple, and undirected. The vertex set and the edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. Given vertices u, v of a graph G, we denote by uv the edge $\{u, v\}$. The *neighborhood* of a vertex v in a graph G is the set $N_G(v)$ of all vertices adjacent to v in G. The *closed neighborhood* of v is the set $N_G(v) \cup \{v\}$, denoted by $N_G[v]$. Given a set $X \subseteq V(G)$, we denote by $N_G(X)$ and $N_G[X]$ the sets $\bigcup_{v\in X} N_G(v) \setminus X$ and $\bigcup_{v\in X} N_G[v]$, respectively. The *degree* of vertex v in G is the cardinality of $N_G(v)$. If the graph G is clear from the context, we simply write $N(v)$, $N[v]$, $N[X]$, and $d(v)$ instead of $N_G(v)$, $N_G[v]$, $N_G[X]$, and $d_G(v)$, respectively. The *codegree* of vertex v in G is the number of vertices in G that are not adjacent to v in G. A vertex v of degree 0 in a graph G is said to be *isolated*, and a vertex v of degree $|V(G)| - 1$ in G is said to be *universal*. Given a graph G, a vertex v is a *pendant* of G if v is adjacent to a single vertex of G. Two vertices u and v are said to be *true twins* (resp., *false twins*) if $N_G[u] = N_G[v]$ (resp., $N_G(u) = N_G(v)$).

A *clique* in a graph G is a set of pairwise adjacent vertices in G, while an *independent set* in a graph G is a set of pairwise non-adjacent vertices in G. A *triangle* in a graph G is a clique of size 3. A set S of vertices in a graph G is a *dominating set* in G if every vertex in G is either in S or has a neighbor in it. A vertex v in a graph G is *simplicial* if its neighborhood is a clique. The *independence number* of G is defined as the cardinality of a largest independent set in G and denoted by $\alpha(G)$. The cardinality of a largest clique in a graph G is the *clique number* of G and we denote it by $\omega(G)$. The *clique covering number* $\theta(G)$ of a graph G is the minimum number of cliques in G needed to cover the vertex set of G. The *chromatic number* $\chi(G)$ of a graph G is the smallest number of colors needed to color the vertices of G so that no two adjacent vertices share the same color, or, equivalently, the minimum number of independent sets in G needed to cover $V(G)$.

2.2 Graph operations and containment relations

A *connected component* (or, simply, a *component*) of a graph G is a maximal connected subgraph of G. The *complement* of a graph G is a graph $\overline{G}$ with vertex set $V(G)$ in which two distinct vertices are adjacent if and only if they are non-adjacent in G. A *coconnected component*, or simply *cocomponent*, of a graph G is the subgraph of G induced by the vertex set of a component of $\overline{G}$. A graph is *coconnected* if its complement is connected. If G is a graph and A and B are disjoint subsets of $V(G)$, we say that they are *complete* (resp., *anticomplete*) *to each other* in G if $\{ab \mid a \in A, b \in B\} \subseteq E(G)$ (resp., $\{ab \mid a \in A, b \in B\} \cap E(G) = \emptyset$). If the vertex set of G can be partitioned into sets V_1 and V_2 that are anticomplete to each other in G, then G is said to be the *disjoint union* of graphs $G[V_1]$ and $G[V_2]$; we denote this by $G = G[V_1]+G[V_2]$. Similarly, if the vertex set of a graph G can be partitioned into two sets V_1 and V_2 that are complete to each other in G, we say that G is the *join* of the subgraphs of G induced by V_1 and V_2; we denote this by $G = G[V_1] * G[V_2]$. Given a non-negative integer k, the disjoint union of k copies of G is denoted by kG.

Given a graph G, its *line graph* is the graph $L(G)$ with vertex set $E(G)$ in which two distinct vertices e and f are adjacent if and only if e and f have a common endpoint as edges in G. A graph F is an *induced subgraph* of a graph G if $V(F) \subseteq V(G)$ and $E(F) = \{uv \in E(G) \mid \{u,v\} \subseteq V(F)\}$; we denote this relation by $F \subseteq_i G$. In this case, graph F will also be called *the subgraph of G induced by $V(F)$* and denoted by $G[V(F)]$. Given a set $S \subseteq V(G)$, we denote by $G - S$ the subgraph of G induced by $V(G) \setminus S$. If F and G are graphs such that no induced subgraph of G is isomorphic to F, we say that G is *F-free*. Given a family $\mathcal{F}$ of graphs, we say that a graph G is *$\mathcal{F}$-free* if no induced subgraph of G is isomorphic to a member of $\mathcal{F}$. *Contracting* an edge $e = uv$ in a graph G is the operation of replacing the vertices u and v in G with a new vertex w that is adjacent precisely to vertices in $(N_G(u) \cup N_G(v)) \setminus \{u,v\}$; the resulting graph is denoted by G/e. A *minor* of a graph G is any graph obtained from G by a sequence of vertex deletions, edge deletions, and edge contractions. An *induced minor* of a graph G is any graph obtained from G by a sequence of vertex deletions and edge contractions. If a graph F is not isomorphic to any minor of G, then G is said to be *F-minor-free*. Similarly, if a graph F is not isomorphic to any induced minor of G, then G is said to be *F-induced-minor-free*; otherwise, we say that G *contains F as an induced minor*. A *subdivision* of a graph G is any graph obtained by repeated application of the operation 'insert a vertex into an edge': replace the edge uv by two edges uw and wv, where w is a new vertex. An *induced topological minor* of a graph G is any graph H such that some subdivision of H is an induced subgraph of G. If a graph F is not isomorphic to any induced minor (resp., induced topological minor) of G, then G is said to be *F-induced-minor-free* (resp., *F-induced-topological-minor-free*). Given a family $\mathcal{F}$ of graphs, we say that a graph G is *$\mathcal{F}$-induced-minor-free* if no induced minor of G is isomorphic to a member of $\mathcal{F}$. If a graph H is an induced topological minor of G, we will often say that G *contains an induced subdivision of H*.

An *induced minor model* of a graph H in a graph G is a collection $\{X_v\}_{v \in V(H)}$ of pairwise disjoint vertex sets $X_v \subseteq V(G)$ such that each induced subgraph $G[X_v]$ is connected, if there is an edge $uv \in E(H)$, then there is an edge between X_u and X_v in G, and for each pair of distinct and non-adjacent vertices $u, v \in V(H)$, $u \neq v$, $uv \notin E(H)$, there are no edges between X_u and X_v. It is not difficult to see that a graph G contains a graph H as an induced minor if and only if there is an induced minor model of H in G. If a graph H is an induced minor of a graph G, we will sometimes refer to the graphs G and H as the *host graph* and the *pattern graph*, respectively.

A *cut partition* of a graph G is a triple (A, B, C) of pairwise disjoint subsets of $V(G)$ such that $A \cup B \cup C = V(G)$, and sets A and B are non-empty and anticomplete to each other. If (A, B, C) is a cut partition of a graph G such that C is a (possibly empty) clique, we say that C is a *clique cutset* in G. A graph is said to be an *atom* if it has no clique cutset. Given a class $\mathcal{G}$ of graphs, we denote by $A(\mathcal{G})$ the class of all atoms that are induced subgraphs of a graph in $\mathcal{G}$. The operation of *gluing* H_1 *and* H_2 *along a clique* produces a graph obtained from $H_1 \cup H_2$ by choosing cliques C_1 in H_1 and C_2 in H_2 such that $|C_1| = |C_2|$, fixing a bijection f from C_1 to C_2, and identifying each vertex $v \in C_1$ with the vertex $f(v)$.

2.3 Particular graphs and graph classes

The complete graph on n vertices is denoted by K_n. We denote by P_n the n-vertex *path*, that is, a graph whose vertex set $\{v_1, \ldots, v_n\}$ can be ordered linearly so that two vertices are adjacent if and only if they appear consecutively in the ordering. Similarly, for an integer $n \geq 3$, we denote by C_n the n-vertex *cycle*, that is, a graph whose vertex set $\{v_1, \ldots, v_n\}$ can be ordered cyclically so that two vertices are adjacent if and only if they appear consecutively in the ordering. A *hole* in a graph G is an induced subgraph isomorphic to the k-vertex cycle C_k for some $k \geq 4$. The complement of a hole in a graph G is called an *antihole* in G. A hole (antihole) is *long* if it has at least five vertices. Holes and antiholes in graphs are called *even* if they have an even number of vertices and *odd* if they have an odd number of vertices. A graph is *chordal* if it contains no holes. We say that a graph G is *weakly chordal* if G contains no long hole, neither long anti-hole. A graph G is *perfect* if every induced subgraph H of G satisfies that $\chi(H) = \omega(H)$, that is, the chromatic number of H equals the size of a maximum clique in H. By the Strong Perfect Graph Theorem [63], perfect graphs are also exactly the *Berge graphs*, that is, (odd hole, odd antihole)-free graphs. A graph is *bipartite* if its vertex set can be partitioned into two independent sets called *parts*. A *complete bipartite* graph is a bipartite graph having all possible edges joining vertices in different parts (in other words, a join of two edgeless graphs). Given two integers $p, q \geq 0$, the graph $K_{p,q}$ is a complete bipartite graph with parts of size p and q, respectively. A *cobipartite* graph is the complement of a bipartite graph, that is, a graph whose vertex set can be partitioned into two cliques. A graph is *acyclic* if it does not contain any cycle. A bipartite graph $G = (A \cup B, E)$ is *biconvex* if it has a biconvex ordering, that is, an ordering of A and B such that for every vertex $a \in A$ (resp. $b \in B$) the neighborhood $N(a)$ (resp. $N(b)$) is an interval of consecutive vertices in the ordering of B (resp. ordering of A). A graph $G = (V, E)$ is a *comparability* graph if it has a transitive orientation, that is, if each of the edges uv of G can be replaced by exactly one of the ordered pairs (u, v) or (v, u) so that the resulting set A of directed edges is transitive (that is, for every three vertices $x, y, z \in V$, if $(x, y) \in A$ and $(y, z) \in A$, then $(x, z) \in A$). A graph G is a *cocomparability* graph if its complement is a comparability graph. *Cographs* are defined as graphs that can be constructed starting from copies of the one-vertex graph using the operations of disjoint union and complementation (see, e.g., [47]). A *rooted tree* is a pair (T, r) where T is a tree and $r \in V(T)$ is the *root* of T. Given two nodes u and v in a rooted tree T, we say that v is a *child* (or *successor*) of u if $uv \in E(T)$ and u belongs to the unique v, r-path in T. A *leaf* of a rooted tree

T is a node without any successors, while an *internal node* of T is a node that is not a leaf.

A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ where T is a tree whose every node t is assigned a vertex subset $X_t \subseteq V(G)$ called a bag such that the following conditions are satisfied:

- Every vertex of G is in at least one bag.
- For every edge $\{u, v\} \in E(G)$ there exists a node $t \in V(T)$ such that X_t contains both u and v.
- For every vertex $u \in V(G)$ the subgraph of T induced by the set $\{t \in V(T) : u \in X_t\}$ is connected (that is, a tree).

The *width* of a tree decomposition $(T, \{X_t\}_{t \in V(T)})$ of a graph G is defined as $\max_{t \in V(T)} |X_t| - 1$. The *treewidth* of a graph G is the minimum possible width of a tree decomposition of G. A graph class $\mathcal{G}$ is said to be of *bounded treewidth* if there exists a nonnegative integer ℓ such that each graph in $\mathcal{G}$ has treewidth at most ℓ.

A *wheel* is a graph that consists of a chordless cycle of length at least four and an additional vertex (called the *center*) that has at least three neighbors in the cycle. A *long wheel* is a wheel with at least 6 vertices. A *twin wheel* is a wheel such that the center has degree three and is adjacent to three consecutive vertices on the cycle. A *broken wheel* is a wheel such that the neighborhood of the center induces a disconnected subgraph of the cycle. A *consecutive wheel* is a wheel such that the neighborhood of the center v induces a connected subgraph of H.

The *diamond* is the graph obtained from the complete graph K_4 by deleting an edge. The *house* is the graph with five vertices a, b, c, d, e and the following edges: ab, bc, cd, de, ae, ad. The *butterfly* is the join of $2P_2$ and P_1. The *gem* is the join of graphs P_1 and P_4. The *claw* is the graph $K_{1,3}$. The *paw* is the graph obtained from the claw by adding to it one edge. The *fork* is the graph obtained from a claw by a single subdivision of one of its edges, that is, the graph with vertex set $\{v_1, v_2, v_3, v_3, v_5\}$ and edge set $\{v_1v_2, v_2v_3, v_3v_4, v_3v_5\}$.

Given three positive integers i, j, k, at most one of which is equal to 1, we denote by $\Gamma_{i,j,k}$ a graph G consisting of two vertices a, b and three paths P, Q, R, each from a to b, and otherwise vertex-disjoint, such that the lengths of the paths P, Q, R are i, j, and k, respectively, and each of the sets $V(P) \cup V(Q)$, $V(P) \cup V(R)$, and $V(Q) \cup V(R)$ induces a cycle in G. We will mostly be interested in the case when the vertices a and b are non-adjacent in $\Gamma_{i,j,k}$, that is, when all the indices i, j, k are at least two. Any such graph will be referred to as a *theta*. Given an integer $k \geq 3$, a *short k-theta* (or simply a *k-theta*) is a graph obtained as the union of k internally disjoint paths of length 3 with common endpoints a and b. More precisely, a k-theta is a graph G with vertex set $V(G) = \{a, a_1, \ldots, a_k, b, b_1, \ldots, b_k\}$, and its set of edges consists of the pairs of the following form: aa_i, bb_i, and a_ib_i for $1 \leq i \leq k$ (see Fig. 2.1). Any graph that is a k-theta for some $k \geq 3$ will be referred to as a *short theta*.

A *prism* is any subdivision of $\overline{C_6}$ in which the two triangles remain unsubdivided; in particular, $\overline{C_6}$ is a prism. For an integer $k \geq 3$, a *short k-prism* (or simply a *k-prism*) is a graph whose vertex set can be partitioned into two n-vertex cliques, say $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$, such that for all $i, j \in \{1, \ldots, k\}$, a_i is adjacent to b_j if and only if $i = j$ (see Fig. 2.1). Any graph that is a k-prism for some $k \geq 3$ will be referred to as a *short prism*.

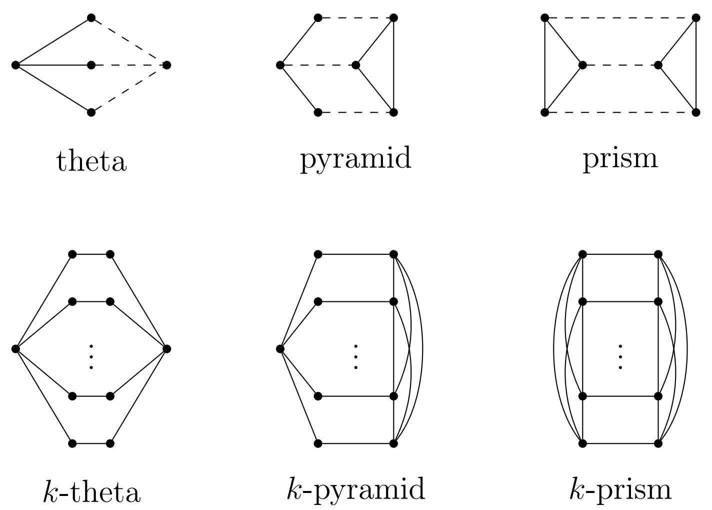


Figure 2.1: Some graphs. Dashed edges are paths of length at least 1.

A *pyramid* is any subdivision of the complete graph K_4 in which one triangle remains unsubdivided, and of the remaining three edges, at least two edges are subdivided at least once. A *k-pyramid* is a graph G with vertex set $V(G) = \{a, a_1, \ldots, a_k, b_1, \ldots, b_k\}$, and with the edge set consisting of the pairs of the following form: aa_i and a_ib_i for $1 \leq i \leq k$, and b_ib_j for $1 \leq i < j \leq k$ (see Fig. 2.1). A 3-path-configuration (or *3PC* for short) is any theta, pyramid, or prism.

2.4 Modular decomposition

Given a graph G and a nonempty set $M \subseteq V(G)$, we say that M is a *module* in G if every vertex not in M is either adjacent to all vertices in M or to none of them. If M_1 and M_2 are two disjoint modules in a graph G, then either G contains all possible edges between M_1 and M_2 in G, or none of them. A module M is *maximal* if $M \subset V(G)$ and there is no module M' in G with $M \subset M' \subset V(G)$. If G and its complement are both connected, then any two maximal modules in G are disjoint; in particular, the set of maximal modules of G forms a partition of $V(G)$. A module M of a graph G is said to be *strong* if for every other module M' in G it holds that either $M \cap M' = \emptyset$, $M \subseteq M'$, or $M' \subseteq M$. A graph G is *prime* if each of its maximal strong modules is a singleton.

Every graph with at least two vertices has a unique partition of its vertex set into maximal strong modules (see, e.g., [132]). If G is disconnected, then the partition is given by the vertex sets of its components; if the complement of G is disconnected, then the partition is given by the vertex sets of its cocomponents. The *representative graph* $R(G)$ of G is any induced subgraph of G obtained by taking an arbitrary but fixed vertex from each maximal strong module of G. Note that the representative graph of G depends on how the vertices from the maximal strong modules are chosen, but any two such graphs are isomorphic to each other, which explains the notation $R(G)$. The representative graph of G is a special case of the following more general construction. Given a graph G and an arbitrary partition $\mathcal{P} = \{M_1, \ldots, M_k\}$ of $V(G)$ into modules of G, we denote by $G/\mathcal{P}$ the corresponding *quotient graph*, which is the induced subgraph of G obtained by taking one vertex from each module $M_j \in \mathcal{P}$.

Partitioning the vertex set of a graph G recursively into maximal strong modules leads to the so-called *modular decomposition* of G, represented with the so-called *modular decomposition tree*. This is a rooted tree T_G such that every node of T_G is labeled with an induced subgraph H_t of G, and every internal node of T_G is of one of the types *parallel*, *series*, or *prime*. The tree T_G is defined recursively as follows.

- If G is the one-vertex graph, then T_G has one node t, labeled with $H_t = G$, and t is the root of T_G.
- Otherwise, T_G is the rooted tree obtained by creating a root node r, labeling the root by the representative graph of G (that is, setting $H_r = R(G)$), and joining the root r with edges to the roots of the modular decomposition trees $T_1, \ldots, T_k$ of the subgraphs of G induced by the maximal strong modules $M_1, \ldots, M_k$ of G. The root node of G is of type *parallel* if G is disconnected, *series* if the complement of G is disconnected, and *prime* if both G and its complement are connected. Each internal node t of T_G with $t \neq r$ belongs to a unique tree T_i and its type in T_G is the same as in T_i.

Given a graph G, the modular decomposition tree T_G of G can be computed in linear time (see [72, 177]). By construction, for every node $t \in V(T_G)$, the subtree of T_G rooted at t is the modular decomposition tree of the subgraph G_t of G induced by the vertices appearing in the one-vertex subgraphs labeling the leaves of this subtree. Furthermore, if the node t is of type prime, then the graph H_t labeling the node is a prime graph.

Part I

PART I: Minimal separators

Chapter 3

Overview and preliminary results

Given two non-adjacent vertices a and b in a graph G, a set $S \subseteq V(G)$ is an (a,b)-separator if a and b are contained in different connected components of $G - S$. If S contains no other (a,b)-separator as a proper subset, then S is a minimal (a,b)-separator. A *minimal separator in G* is a set $S \subseteq V(G)$ that is a minimal (a,b)-separator for some pair of non-adjacent vertices a and b. A graph class $\mathcal{G}$ is *tame* if there exists a polynomial $p : \mathbb{R} \to \mathbb{R}$ such that for every graph $G \in \mathcal{G}$, we have $s(G) \leq p(|V(G)|)$, where $s(G)$ denotes the number of minimal separators in G. A graph class is *feral* if there exists a constant $c > 1$ so that for arbitrarily large n there is an n-vertex graph in the class with at least c^n minimal separators.

3.1 Overview

Minimal separators in graphs were studied since at least 1960s when Dirac [95] characterized chordal graphs as graphs whose minimal separators are cliques. The nice structure and properties of chordal graphs that allow for efficient algorithms to many problems that are NP-hard in general motivated the various studies of this class (see, e.g., [127]). Widely studied problems on graphs are concerned with computing an embedding of an arbitrary graph into a chordal graph with various properties (see, e.g., [10, 232]). In particular, one can add edges to any given graph so that the resulting graph, called a *triangulation* of the input graph, is chordal. Clearly, there are many different triangulations for a given graph in general, so various parameters of triangulations can be minimized. Known graph problems related to triangulations of graphs are called Minimum Fill-in and Treewidth and both of them are known to be NP-hard even for the class of cobipartite graphs [10, 232]. The Minimum Fill-in asks to find a triangulation with the fewest number of edges, while the Treewidth asks to find a triangulation where the size of the largest clique is minimized. Kloks et al. in 1993 [150] indicated a strong connection between the minimal separators of a graph and the solutions to both of these problems. In particular, they claimed that both problems Minimum Fill-in and Treewidth are solvable in polynomial time when restricted to any class of graphs having a polynomial-time algorithm that computes the set of all minimal separators for every graph in the class. It turned out that their proof was not correct [151], so it remained an open question whether Treewidth and Minimum Fill-in are tractable in polynomial time for every graph class having an algorithm that enumerates all minimal separators in polynomial time. Note that the first algorithm that efficiently enumerates all minimal separators in a graph was given by

Kloks and Kratsch, in 1994 [152], and this initiated the more general study of classes of graphs having polynomially many minimal separators. Parra and Scheffler [193] showed that minimal triangulations of a graph can be obtained by turning into cliques a maximal set of *pairwise parallel*[1] minimal separators in the input graph. This was the first construction of minimal triangulations of a graph that did not rely on vertex eliminations but rather on minimal separators. The revised conjecture from [150] that TREEWIDTH and MINIMUM FILL-IN are tractable in polynomial time for every graph class having a polynomial number of minimal separators was proved by Bouchitté and Todinca in [42] using the concept of potential maximal cliques, which they introduced in previous work [39, 40]. A *potential maximal clique* in a graph is a maximal clique in some minimal triangulation of the graph. Bouchitté and Todinca showed that the number of potential maximal cliques is polynomially bounded by the number of minimal separators and that it is possible to enumerate the potential maximal cliques in time polynomial in their number [42]. Consequently, TREEWIDTH and MINIMUM FILL-IN were shown to be polynomially tractable for tame classes of graphs.

During the period of intensive studies of minimal separators done by Bouchitté and Todinca [39, 40, 41, 42], some other related independent studies appeared. In particular, Berry and Bordat [24] generalized the characterizations of chordal graphs obtained by Dirac [95] to arbitrary graphs via moplexes. Connections between minimal triangulations and minimal separators of a graph were further studied by Fomin and Villanger [112] when they showed that the problem of finding a maximum induced subgraph of treewidth at most a constant t can be solved in polynomial time for classes of graphs with a polynomially bounded number of minimal separators. This result was then generalized by Fomin, Todinca, and Villanger [111] who gave a general framework describing NP-hard problems that are solvable in polynomial time in any tame graph class. In particular, building on connections between minimal separators and potential maximal cliques, they proved the following result. We denote by $\mathsf{tw}(G)$ the treewidth of a graph G. For a fixed integer $t \geq 0$ and a fixed CMSO_2 formula ϕ, consider the following computational problem.

(t, ϕ)-MAXIMUM WEIGHT INDUCED SUBGRAPH
Input: A graph G equipped with a vertex weight function $w\colon V(G) \to \mathbb{Q}_+$.
Task: Find a set $X \subseteq V(G)$ of maximum possible weight such that $G[X] \models \phi$ and $\mathsf{tw}(G[X]) \leq t$, or conclude that no such set exists.

Theorem 3.1.1 (Fomin, Todinca, and Villanger [111])**.** *For a fixed integer $t \geq 0$, fixed CMSO_2 formula ϕ, and any tame graph class $\mathcal{G}$, the (t, ϕ)-*MAXIMUM WEIGHT INDUCED SUBGRAPH *problem is solvable in polynomial time for graphs in $\mathcal{G}$.*

Let us remark that the algorithm given by Theorem 3.1.1 is *robust* in the sense of Raghavan and Spinrad [208], that is, it produces the correct output regardless of whether the input actually belongs to the restricted domain or not. The algorithm works on any graph G and correctly solves the problem whenever the number of minimal separators is bounded by a polynomial (which is, in particular, the case when $G \in \mathcal{G}$); otherwise, the algorithm correctly reports that the given graph does not belong to $\mathcal{G}$. Examples of problems captured by the above framework include MAXIMUM WEIGHT

[1] Two separators are called *parallel* if none of them contains two vertices that are separated by the other.

Independent Set, Maximum Weight Induced Matching, Maximum Weight Induced Forest (which is equivalent, by complementation, to Minimum Weight Feedback Vertex Set), Longest Induced Path, and many others (see [111]). Minimal separators have also been studied in the recent literature from various other points of view (see, e.g., [3, 129, 133, 144, 156, 179]).

The above algorithmic results motivate the quest of identifying tame graph classes and classifying them when restricted to particular families of graphs. Known tame graph classes include chordal graphs [212] and their generalization weakly chordal graphs [41], permutation graphs [32, 148] and more generally cocomparability graphs of bounded interval dimension [92], circular-arc graphs [154], circle graphs [149], polygon circle graphs [221], distance-hereditary graphs [150], probe interval graphs [55], AT-free, co-AT-free graphs [155], P_4-sparse graphs [187], extended P_4-laden graphs [196], and graphs with minimal separators of bounded size [218].

The problem of computing the number of minimal separators of a disconnected graph can be reduced to the same problem on each component, and similarly for graphs whose complements are disconnected (see, e.g., [195]). We examine the consequences of these results for tame graph classes and show that the problem of determining if a hereditary graph class $\mathcal{G}$ is tame can be reduced to the same problem on the subclass of $\mathcal{G}$ obtained by certain operations on graphs (for example: join, disjoint union, gluing along cliques, etc.). We use the above results, along with constructions of graph classes that lack the property of tameness, to completely characterize which graph classes defined by forbidden induced subgraphs with at most four vertices are tame. Furthermore, building on recent works of Gartland and Lokshtanov [121], and of Gajarský et al. [115], we show that every graph class defined by a single forbidden induced minor or induced topological minor is either tame or feral, and classify the two cases. We provide polynomial-time recognition algorithms for the maximal tame graph classes obtained in the above characterizations by a single forbidden induced minor or induced topological minor.

Among the tame classes of graphs, of special interest are classes having a linear number of minimal separators, making the algorithms based on minimal separators particularly efficient. For example, n-vertex split graphs have no more than n minimal separators [196] and, more generally, the same is true for n-vertex chordal graphs [212] and $2P_2$-free graphs. In the doctoral thesis we address the extremal question of determining the maximum number of minimal separators in an n-vertex graph from a given class, for a number of interrelated graph classes with at most a linear number of minimal separators.

Secondly, we generalize the concept of chordal graphs by investigating classes of graphs for which every minimal separator of every graph in the class is a union of k cliques, for some fixed nonnegative integer k. In the doctoral thesis we consider a more general concept and given a class $\mathcal{C}$ of graphs, we denote by $\mathcal{G}_{\mathcal{C}}$ the class of all graphs G such that every minimal separator of G induces a graph from $\mathcal{C}$. Since complete graphs have no separators, we see that for all classes $\mathcal{C}$, the class $\mathcal{G}_{\mathcal{C}}$ contains all complete graphs. For a nonnegative integer k, we denote by $\mathcal{G}_k$ the class of all graphs G that have the property that every minimal separator of G is k-simplicial, that is, a union of k (possibly empty) cliques. If $k = 1$ (resp., $k = 2$), we simplify our terminology, so that k-simplicial becomes *simplicial* (resp., *bisimplicial*). Obviously, $\mathcal{G}_0 \subseteq \mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \ldots$. Recall that chordal graphs are precisely the graphs with minimal separators that are cliques, so they actually correspond to the case when $k = 1$ [95]. If $k = 0$, then we obtain the class of complete graphs.

We study the classes of the form $\mathcal{G}_{\mathcal{C}}$, where $\mathcal{C}$ is a hereditary class, that is, a class of graphs closed under vertex deletion. We place particular emphasis on the classes $\mathcal{G}_k$, and, in particular, $\mathcal{G}_2$. By the above, the class $\mathcal{G}_2$ is the class of graphs with bisimplicial minimal separators, and $\mathcal{G}_2$ contains all chordal graphs. Moreover, it is easy to see that all circular-arc graphs (that is, intersection graphs of arcs on a circle) belong to $\mathcal{G}_2$. This motivates the study of computational complexities of various graph problems in $\mathcal{G}_k$, $k \geq 2$, as well as the structure of graphs belonging to $\mathcal{G}_k$, with particular emphasis on the case $k = 2$.

Throughout the Part I of this thesis we denote by $\mathcal{S}_G(a, b)$ the set of all minimal (a, b)-separators in a given graph G and by $\mathcal{S}_G$ the set of all minimal separators in G. The cardinality of $\mathcal{S}_G$ will be denoted by $s(G)$.

Given a graph G and a set $S \subseteq V(G)$, a component C of the graph $G - S$ is said to be S-*full* if every vertex in S has a neighbor in C, or, equivalently, if $N_G(V(C)) = S$. The following well-known lemma characterizes minimal separators (see, e.g., [127]).

Lemma 3.1.2. *Given a graph $G = (V, E)$, a set $S \subseteq V$ is a minimal separator in G if and only if the graph $G - S$ contains at least two S-full components.*

Our proofs will make use of some preliminary results. Among them is the structure of minimal separators in a $2P_2$-free graph, given by the following lemma.

Lemma 3.1.3. *Let G be a $2P_2$-free graph and let S be a minimal separator in G. Then there exists a vertex $v \in V(G)$ such that $S = N(v)$.*

Proof. By Lemma 3.1.2, the graph $G - S$ has two S-full components C and D. If both C and D have at least two vertices, then each of them contains at least one edge. Since C and D are anticomplete to each other, these edges form a $2P_2$ in G. We may thus assume, by symmetry, that $C = \{v\}$ for some $v \in V(G)$. Then it follows that $N(v) \subseteq S$ and since every vertex of S is adjacent to v, we must have $S = N(v)$, as claimed. □

3.2 Graph operations and their influence on the number of minimal separators

In this section, we study the effect of various graph operations on the number of minimal separators. The family of minimal separators of a disconnected graph can be computed from the families of minimal separators of its components, and a similar statement holds for graphs whose complements are disconnected. The correspondences are as follows, see Pedrotti and de Mello [196].

Theorem 3.2.1. *If G is a disconnected graph, with components $G_i, \ldots, G_k$, then $\mathcal{S}_G = \{\emptyset\} \cup \bigcup_{i=1}^{k} \mathcal{S}_{G_i}$. If G is the join of graphs $G_1, \ldots, G_k$, then $S \in \mathcal{S}_G$ if and only if there exists some $i \in \{1, \ldots, k\}$ and some $S_i \in \mathcal{S}_{G_i}$ such that $S = S_i \cup (V(G) \setminus V(G_i))$.*

Using Theorem 3.2.1 we can derive the formulas for the number of minimal separators and their corollaries.

Corollary 3.2.2. *Let G be a disconnected graph, with components $G_1, \ldots, G_k$. Then $s(G) = \sum_{i=1}^{k} s(G_i) + 1$.*

Proof. Immediate from the first statement of Theorem 3.2.1 and the fact that sets $\{\emptyset\}, \mathcal{S}_{G_1}, \ldots, \mathcal{S}_{G_k}$, $i \in \{1, \ldots, k\}$ are pairwise disjoint. □

Corollary 3.2.3. *Let $G_1, \ldots, G_k$ be graphs and let G be the join of $G_1, \ldots, G_k$. Then $s(G) = \sum_{i=1}^{k} s(G_i)$.*

Proof. From Theorem 3.2.1 we have that $S \in \mathcal{S}_G$ if and only if there exists some $i \in \{1, \ldots, k\}$ and some $S_i \in \mathcal{S}_{G_i}$ such that $S = S_i \cup (V(G) \setminus V(G_i))$. Clearly, if $i, j \in \{1, \ldots, k\}$, $i \neq j$, then $S_i \cup (V(G) \setminus V(G_i)) \neq S_j \cup (V(G) \setminus V(G_j))$ and the sets $\{S_i \cup (V(G) \setminus V(G_i)) \mid S_i \in \mathcal{S}_{G_i}\}$ and $\{S_j \cup (V(G) \setminus V(G_j)) \mid S_j \in \mathcal{S}_{G_j}\}$ are disjoint. Moreover, if for some $i \in \{1, \ldots, k\}$ sets S_i and S_i' are distinct minimal separators in G_i, then also the sets $S_i \cup (V(G) \setminus V(G_i))$ and $S_i' \cup (V(G) \setminus V(G_i))$ are distinct. Therefore $|\{S_i \cup (V(G) \setminus V(G_i)) \mid S_i \in \mathcal{S}_{G_i}\}| = s(G_i)$ for all $i \in \{1, \ldots, k\}$. It follows that $s(G) = \sum_{i=1}^{k} s(G_i)$, as claimed. □

Corollary 3.2.3 implies the following result for graphs with disconnected complements.

Corollary 3.2.4. *Let G be a graph whose complement is disconnected, with cocomponents $G_1, \ldots, G_k$. Then $s(G) = \sum_{i=1}^{k} s(G_i)$.*

The following lemma gives an upper bound on the number of minimal separators of a graph with a clique cutset in terms of the number of minimal separators of two smaller graphs. In the proof we use the following notation: given a graph G, a set $S \subseteq V(G)$, and a vertex $x \in V(G) \setminus S$, we denote by $\Gamma_{G,S,x}$ the component of $G - S$ containing x.

Lemma 3.2.5. *Let G be a graph that admits a cut-partition (A, B, C) such that C is a clique. Then $s(G) \leq s(G[A \cup C]) + s(G[B \cup C]) + 1$.*

Proof. Denote by G_1 (resp. G_2) the subgraph of G induced by $A \cup C$ (resp. $B \cup C$). We show the claimed inequality by proving that $\mathcal{S}_G \subseteq \mathcal{S}_{G_1} \cup \mathcal{S}_{G_2} \cup \{C\}$. Suppose S is a minimal (x, y)-separator in G and $S \neq C$. The fact that C is a clique implies that either $C \cap V(\Gamma_{G,S,x}) = \emptyset$ or $C \cap V(\Gamma_{G,S,y}) = \emptyset$. By symmetry, we may assume that $C \cap V(\Gamma_{G,S,x}) = \emptyset$ and furthermore that $V(\Gamma_{G,S,x}) \subseteq A$. Since A is anticomplete to B, this implies that $S \cap B = \emptyset$ and thus $S \subseteq V(G_1)$. We complete the proof by showing that S is a minimal separator in G_1.

Suppose first that $S \subseteq C$. Since $S \neq C$, there exists a vertex $z \in C \setminus S$. We claim that S is a minimal (x, z)-separator in G_1. Since $V(\Gamma_{G,S,x}) \subseteq A$, vertices x and z are separated in $G_1 - S$. From the minimality of S it follows that $\Gamma_{G,S,x}$ is an S-full component of $G_1 - S$. Furthermore, since $S \subseteq C$ and C is a clique in G_1, every vertex in S is adjacent to z in G_1. It follows that S is a minimal (x, z)-separator in G_1, as claimed.

We may thus assume that $S \nsubseteq C$ and hence S contains a vertex from A. Since A is anticomplete to B and $\Gamma_{G,S,y}$ is an S-full component of $G - S$, this component cannot be entirely contained in B. Let z be a vertex in $V(\Gamma_{G,S,y}) \cap V(G_1)$. We claim that S is a minimal (x, z)-separator in G_1. Since G_1 is an induced subgraph of G and S separates x from z in G, we infer that S is an (x, z)-separator in G_1. Note that the components of $G_1 - S$ containing x and z, respectively, are $\Gamma_{G,S,x}$ and the subgraph of G_1 induced by $V(\Gamma_{G,S,y}) \cap V(G_1)$. Thus, every vertex in S is adjacent in G_1 to a vertex in $\Gamma_{G_1,S,x}$. It remains to show that every vertex in S is adjacent in G_1 to a vertex in $\Gamma_{G_1,S,z}$. If $V(\Gamma_{G,S,y}) \subseteq V(G_1)$, then $\Gamma_{G_1,S,z} = \Gamma_{G,S,y}$ and the conclusion is clear. Otherwise, $V(\Gamma_{G,S,y}) \cap B \neq \emptyset$, which implies that $V(\Gamma_{G,S,y}) \cap C \neq \emptyset$. Since $S \subseteq A \cup C$ and A is anticomplete to B, every vertex of S that is adjacent in G to a vertex in $V(\Gamma_{G,S,y}) \cap B$ belongs to C and is therefore adjacent in G_1 also to a vertex in $V(\Gamma_{G,S,y}) \cap C \subseteq V(\Gamma_{G_1,S,z})$. Therefore, using the fact that every vertex in S is adjacent in G to a vertex in $\Gamma_{G,S,y}$ we infer that every vertex in S is adjacent in G_1 to a vertex in $\Gamma_{G_1,S,z}$, as claimed. □

Recall now the definitions related to modular decomposition, from Section 2.4. As shown by Pedrotti and de Mello [196], the set of minimal separators in a graph G can be computed from the sets of minimal separators of its representative graph and of its subgraphs induced by the proper maximal strong modules. The correspondence is as follows.

Theorem 3.2.6. *Let G be a graph, let $\{M_1, \ldots, M_k\}$ be a partition of $V(G)$ into proper maximal strong modules, and let G' be the representative graph of G. For each $i \in \{1, \ldots, k\}$, let v_i be the representative vertex of module M_i. Then $\mathcal{S}_G = \mathcal{S}_1 \cup \mathcal{S}_2$, where*

1. $\mathcal{S}_1 = \{S \cup M_{i_1} \cup M_{i_2} \cup \cdots \cup M_{i_j} \mid q \in \{1, \ldots, k\}, S \in \mathcal{S}_{G[M_q]}, N_{G'}(v_q) = \{v_{i_1}, \ldots, v_{i_j}\}\}$,
2. $\mathcal{S}_2 = \{M_{i_1} \cup \ldots \cup M_{i_j} \mid \{v_{i_1}, v_{i_2}, \ldots, v_{i_j}\} \in \mathcal{S}_{G'}\}$.

Theorem 3.2.6 can be used to express the number of minimal separators in a graph G in terms of the number of minimal separators of its representative graph and of its subgraphs induced by the proper maximal strong modules.

Proposition 3.2.7. *Let G be a graph, let $\{V_1, \ldots, V_k\}$ be a partition of $V(G)$ into modules, and let G' be the corresponding quotient graph. Then*

$$s(G) = \sum_{i=1}^{k} s(G[V_i]) + s(G').$$

Proof. Let $V(G') = \{v_1, \ldots, v_k\}$ with $v_i \in V_i$ for $i \in \{1, \ldots, k\}$. Let us also denote $G_i = G[V_i]$ for $i \in \{1, \ldots, k\}$. We will prove the proposition by constructing a partition of the set of minimal separators in G and then defining bijective functions between each of the sets in the partition of $\mathcal{S}_G$ and the sets of minimal separators in $G_1, \ldots, G_k$, and G', respectively.

For every minimal separator S in G we fix two S-full components of $G - S$, call them C_S and D_S. First we define a partition of the set $\mathcal{S}_G$ into the sets $\mathcal{S}'_G$ and $\mathcal{S}^i_G$, $i = 1, \ldots, k$ as follows:

$$\mathcal{S}^i_G = \{S \in \mathcal{S}_G \mid V(C_S) \cap V_i \neq \emptyset \wedge V(D_S) \cap V_i \neq \emptyset\} \tag{3.1}$$

$$\mathcal{S}'_G = \mathcal{S}_G \setminus \left(\bigcup_{i=1}^{k} \mathcal{S}^i_G \right). \tag{3.2}$$

From the definition of sets $\mathcal{S}_G^i$ and $\mathcal{S}_G'$ it follows that the union of these sets is $\mathcal{S}_G$. We still have to prove that they are pairwise disjoint. The set $\mathcal{S}_G'$ is clearly disjoint from the other sets. Let $i, j \in \{1, \ldots, k\}, i \neq j$. We want to prove that $\mathcal{S}_G^i \cap \mathcal{S}_G^j = \emptyset$. Suppose for a contradiction that there is some minimal separator $S \in \mathcal{S}_G^i \cap \mathcal{S}_G^j$. Then it holds that $V(C_S) \cap V_i \neq \emptyset$, $V(C_S) \cap V_j \neq \emptyset$, $V(D_S) \cap V_i \neq \emptyset$, and $V(D_S) \cap V_j \neq \emptyset$. It is not difficult to see that the vertices v_i and v_j corresponding to the modules V_i and V_j are non-adjacent in G', since otherwise modules V_i and V_j would be fully adjacent in G, in contradiction with the fact that $V(C_S)$ and $V(D_S)$ are anticomplete to each other in G. Since C_S is connected, there is a path $P = u_1 \ldots u_m$ in C_S connecting a vertex in $V_i \cap V(C_S)$ with a vertex in $V_j \cap (C_S)$. Since modules V_i and V_j are anticomplete to each other, every such path goes through some other modules in G. Moreover, since between every two modules there are either all edges or none of them, it follows that the path obtained from P by replacing vertex u_1 with a vertex from $V_i \cap V(D_S)$ connects C_S and D_S in $G - S$, implying that these two components are not separated in $G - S$. This is a contradiction, which shows that sets $\mathcal{S}_G^i$ and $\mathcal{S}_G^j$ are disjoint. Therefore, equation (3.1) indeed defines a partition of $\mathcal{S}_G$.

Now we will define the bijective functions, as follows:

$$\phi_i : \mathcal{S}_G^i \to \mathcal{S}_{G_i}, \quad \phi_i(S) = S \cap V_i, \text{ for all } S \in \mathcal{S}_G^i, \tag{3.3}$$
$$\phi' : \mathcal{S}_G' \to \mathcal{S}_{G'}, \quad \phi'(S) = \{v_j \mid V_j \subseteq S\}, \text{ for all } S \in \mathcal{S}_G'. \tag{3.4}$$

First we will prove that the image of every function ϕ_i or ϕ' is a minimal separator in G_i or G', respectively. After that, we will prove that each of these functions is bijective.

Let $i \in \{1, \ldots, k\}$ and let $S \in \mathcal{S}_G^i$. Since (3.1) defines a partition, it follows that $V(C_S) \subseteq V_i$ and $V(D_S) \subseteq V_i$. As every vertex of S has neighbors in both C_S and D_S in G and since $\phi_i(S) \subseteq S$, it follows that C_S and D_S are $\phi_i(S)$-full components of $G_i \setminus \phi_i(S)$, implying that $\phi_i(S) \in \mathcal{S}_{G_i}$. Let now $S \in \mathcal{S}_G'$. We want to prove that $\phi'(S) \in \mathcal{S}_{G'}$. The definition of $\mathcal{S}_G'$ implies that $C_S \subseteq V_i$ and $D_S \subseteq V_j$ for a pair of distinct indices $i, j \in \{1, \ldots, k\}$. Recall that every vertex in S has neighbors in C_S and in D_S. Thus, since any two modules from $\{V_1, \ldots, V_k\}$ are either complete to each other or anticomplete to each other, we infer that every module V_q that contains a vertex from S is fully contained in S (as otherwise S would not separate C_S and D_S in G). Since C_S is S-full, for every $q \in \{1, \ldots, k\}$ such that $V_q \subseteq S$, it holds that v_q is adjacent to v_i in G'. Similarly v_q is adjacent to v_j in G'. This implies that the components of $G' - \phi'(S)$ containing v_i and v_j, respectively, are $\phi'(S)$-full and thus $\phi'(S) \in \mathcal{S}_{G'}$.

Now we have to prove that these functions are bijective. To do that, we show that each of them has an inverse. To this end, we define for each $i \in \{1, \ldots, k\}$, a function $\psi_i : \mathcal{S}_{G_i} \to \mathcal{S}_G^i$ as $\psi_i(S) = S \cup \left(\bigcup_{v_j \in N_{G'}(v_i)} V_j\right)$. Moreover, we define $\psi' : \mathcal{S}_{G'} \to \mathcal{S}_G'$ as $\psi'(S) = \bigcup_{v_j \in S} V_j$. From the definition of these functions it is clear that ψ_i is the inverse of ϕ_i, for all $i \in \{1, \ldots, k\}$ and that ψ' is the inverse of ϕ'.

Let $i \in \{1, \ldots, k\}$ and let $S \in \mathcal{S}_{G_i}$. We want to prove that $\psi_i(S) \in \mathcal{S}_G^i$. Since S is minimal separator in G_i, there are S-full components of $G_i - S$, say C and D. Observe that $N_G(C) \subseteq \psi(S)$ and $N_G(D) \subseteq \psi(S)$, by definition of $\psi(S)$. So it has to be that there is no path connecting components C and D in $G - \psi_i(S)$. Since for every vertex $x \in \bigcup_{v_j \in N_{G'}(v_i)} V_j$ it holds that x has neighbors in S full components C

and D, we have that C and D are $\psi_i(S)$-full components of G, inferring that $\psi_i(S)$ is a minimal separator in G, with $V(C) \cap \psi_i(S) \neq \emptyset$ and $V(D) \cap \psi_i(S) \neq \emptyset$, so it follows that $\psi_i(S) \in \mathcal{S}_G^i$. In order to prove that ϕ_i and ϕ_i are inverse functions, we can simply show that $\phi_i(\psi_i(S) = S$ for every $S \in \mathcal{S}_{G_i}$. From definitions of these two functions we obtain $\phi_i(\psi_i(S)) = \phi_i\left(S \cup \left(\bigcup_{v_j \in N_{G'}(v_i)} V_j\right)\right) = S$, as we wanted to prove.

Let now $S \in \mathcal{S}_{G'}$ and let C and D be two S-full components of $G-S$. Let us define the sets $C' = \bigcup_{v_j \in C} V_j$ and $D' = \bigcup_{v_j \in D} V_j$. Clearly $C' \cap D' = \emptyset$. We claim that C' and D' are $\psi'(S)$-full components of $G - \psi'(S)$. If there exist a path connecting C' and D' in $G - \psi'(S)$, let us denote with P the shortest such path. Then the path in G' consisting of vertices $\{v_i \mid V_i \cap P \neq \emptyset\}$ connects C and D in $G' - S$; a contradiction. Thus the set $\psi'(S)$ separates components C' and D' in G. By definition, if two vertices in G' are adjacent, then the corresponding modules in G are complete to each other. So it follows that every vertex in $\psi'(S)$ has neighbors in both components C' and D', inferring that these two components are $\psi'(S)$-full, as claimed. In order to show that ψ' and ϕ' are inverse functions, it suffices to show that $\phi'(\psi'(S)) = S$, what is clear from the definition of functions. Since the set of domains of the proposed function is the partition of the set $\mathcal{S}_G$, it follows that the cardinality of $\mathcal{S}_G$ is equal to the sum of cardinalities of the functions' domains. As all functions defined above are bijective, it follows that

$$s(G) = \sum_{i=1}^{k} |\mathcal{S}_G^i| + |\mathcal{S}_G'| = \sum_{i=1}^{k} |\mathcal{S}_{G_i}| + |\mathcal{S}_{G'}| = \sum_{i=1}^{k} s(G_i) + s(G'),$$

as claimed. □

Vertex deletions

In this section we describe the number of minimal separators in a graph obtained by the deletion of a vertex from the initial graph. Recall that in Lemma 3.1.2 a minimal separator in a graph G was characterized as a set $S \subseteq V(G)$ satisfying that the graph $G - S$ contains at least two S-full components. This brings the following corollary.

Corollary 3.2.8. *Let S be a minimal separator in a graph G. Then for every $v \in S$ the set $S \setminus \{v\}$ is a minimal separator in $G - v$.*

Proof. Let $G' = G - v$ and $S' = S \setminus \{v\}$. Since S is a minimal separator in G, there exist two S-full components C and D in $G - S$. Since $G - S = G' - S'$ and $S' \subseteq S$, it follows that C and D are also S'-full components of $G' - S'$. Hence, S' is a minimal separator in G'. □

Corollary 3.2.9. *Let S be a minimal separator in a graph G. Then for every $S^* \subseteq S$ the set $S \setminus S^*$ is a minimal separator in $G - S^*$.*

McKee observed in [178] that if G_1 is an induced subgraph of G_2, then every minimal separator of G_1 is contained in a minimal separator of G_2. The proof actually shows that the number of minimal separators is monotone under vertex deletion.

Proposition 3.2.10. *If G_1 is an induced subgraph of G_2, then $s(G_1) \leq s(G_2)$.*

Given a graph G and a vertex $v \in V(G)$, it follows that $s(G-v) \leq s(G)$. The following results gives lower-bound on $s(G-v)$ in terms of $s(G)$, for a particular choice of vertex v. First we consider the case when v is a universal vertex in G.

Proposition 3.2.11. *Let G be a graph with at least two vertices and let v be a universal vertex in G. Then $s(G) = s(G-v)$.*

Proof. Immediate from Corollary 3.2.3, using the fact that G is isomorphic to the join of $G-v$ and K_1, and that $s(K_1) = 0$. □

Of particular interest are vertices that have the same neighborhoods in a graph, and the following lemma is an auxiliary result in this direction.

Lemma 3.2.12. *Let G be a graph and v, w vertices in G such that $N_G(v) \setminus \{w\} = N_G(w) \setminus \{v\}$. If S is a minimal separator in G, then $v \in S$ if and only if $w \in S$.*

Proof. Let $S \in \mathcal{S}_G$. By symmetry, it suffices to show that $v \in S$ implies $w \in S$. Suppose for a contradiction that $v \in S$ but $w \notin S$. By Lemma 3.1.2, the graph $G-S$ has two S-full components, say C and D. We may assume without loss of generality that $w \notin V(C)$. Since C is an S-full component of $G-S$, vertex $v \in S$ has a neighbor y in $V(C)$. But now, y is a vertex contained in $N_G(v) \setminus \{w\}$ but not in $N_G(w) \setminus \{v\}$, a contradiction. □

Recall that two vertices u and v in a graph G are said to be true twins (resp., false twins) if $N_G[u] = N_G[v]$ (resp., $N_G(u) = N_G(v)$).

Proposition 3.2.13. *Let G be a graph having a pair of true twins v, w with $v \neq w$. Then $s(G) = s(G-v)$.*

Proof. Let $G' = G - v$. From Proposition 3.2.10 it follows that $s(G) \geq s(G-v)$, so we have to prove that $s(G) \leq s(G-v)$. We will prove it by exhibiting a one-to-one mapping ϕ from $\mathcal{S}_G$ to $\mathcal{S}_{G'}$. The function is defined by the following rule: for every $S \in \mathcal{S}_G$, we set $\phi(S) = S \setminus \{v\}$.

We first show that ϕ maps minimal separators in G to minimal separators in G'. Let $S \in \mathcal{S}_G$. If $v \in S$, then $\phi(S) = S \setminus \{v\}$, which is a minimal separator in G' by Corollary 3.2.8. Suppose now that $v \notin S$. Then $\phi(S) = S$ and $w \notin S$ by Lemma 3.2.12. Let C and D be two S-full components of $G-S$ and let K be the component of $G-S$ containing v. If $K \notin \{C, D\}$, then C and D are two S-full components of $G'-S$ and hence $\phi(S) = S \in \mathcal{S}_{G'}$ in this case. We may thus assume that $K = C$. Note that $w \in V(C)$ since vertices v and w are adjacent in G. Moreover, since v and w are true twins in G, they are also true twins in C. This implies that the graph $C - v$ is connected and hence a component of $G' - S$. Note that D is an S-full component of $G' - S$. We complete the proof that S is a minimal separator in G' by showing that $C - v$ is also an S-full component of $G' - S$. Consider an arbitrary vertex $x \in S$. Since C is an S-full component of $G - S$, vertex x has a neighbor in C. However, since v and w are true twins in G, it cannot be that $N_G(x) \cap V(C) = \{v\}$. It follows that x also has a neighbor in $C - v$. Therefore, $C - v$ is an S-full component of $G' - S$, as claimed.

It remains to show that ϕ is one-to-one. Let S_1 and S_2 be distinct minimal separators in G. If $S_1 \cap \{v\} \neq S_2 \cap \{v\}$, then we have $\phi(S_1) \cap \{w\} \neq \phi(S_1) \cap \{w\}$, inferring that

$\phi(S_1) \neq \phi(S_2)$. If none of the sets S_1, S_2 contains v, then $\phi_1(S_1) = S_1$ and $\phi(S_2) = S_2$. If both of the sets S_1 and S_2 contain v, then $\phi(S_1) = S_1 \setminus \{v\} \neq S_2 \setminus \{v\} = \phi(S_2)$. It follows that ϕ is one-to-one, as claimed. □

Proposition 3.2.14. *Let G be a graph having a pair of false twins v, w with $v \neq w$. Then*

$$s(G - v) \leq s(G) \leq s(G - v) + 1\,.$$

Proof. Let $G' = G - v$. The first inequality follows directly from Proposition 3.2.10. In order to prove the second inequality, we use the following notation: $\mathcal{S}_G^- = \mathcal{S}_G \setminus \{N_G(v)\}$ and $\mathcal{S}_{G'}^- = \mathcal{S}_{G'} \setminus \{N_G(v)\}$. Observe that $\{v\}$ and $\{w\}$ are $N_G(v)$-full components of $G - N_G(v)$, hence by Lemma 3.1.2, we have $N_G(v) \in \mathcal{S}_G$. It follows that $|\mathcal{S}_G^-| = s(G) - 1$ and that $|\mathcal{S}_{G'}^-| \in \{s(G'), s(G') - 1\}$, depending on whether $N_G(v) \in \mathcal{S}_{G'}$ or not. We will prove that $s(G) \leq s(G - v) + 1$ by exhibiting a one-to-one mapping ϕ from $\mathcal{S}_G^-$ to $\mathcal{S}_{G'}^-$. This will suffice, as it will imply $|\mathcal{S}_G^-| + 1 \leq |\mathcal{S}_{G'}^-| + 1$, which together with $|\mathcal{S}_{G'}^-| + 1 \leq s(G - v) + 1$ implies the desired inequality.

The mapping ϕ is defined by the following rule: for every $S \in \mathcal{S}_G^-$, we set $\phi(S) = S \setminus \{v\}$. We first show that for every $S \in \mathcal{S}_G^-$, its image under ϕ is a minimal separator in G'. Let $S \in \mathcal{S}_G^-$. Then $S \neq N_G(v)$. If $v \in S$, then $\phi(S) = S \setminus \{v\}$, which is a minimal separator in G' by Corollary 3.2.8. Moreover, it is not possible that $\phi(S) = N_G(v)$, since that would imply $N_G(v) \subseteq S$ and no component of $G - S$ could be S-full. Thus, if $v \in S$, then $\phi(S) = S \setminus \{v\} \in \mathcal{S}_{G'}^-$. Suppose now that $v \notin S$. Then $\phi(S) = S$, which implies $\phi(S) \neq N_G(v)$. By Lemma 3.2.12, we have $w \notin S$. Let C and D be two S-full components of $G - S$ and let K be the component of $G - S$ containing v. If $K \notin \{C, D\}$, then C and D are two S-full components of $G' - S$ and hence $\phi(S) = S \in \mathcal{S}_{G'}^-$ in this case. We may thus assume that $K = C$. If $C = \{v\}$, then $S = N_G(v)$ and we have a contradiction. So it follows that C has more than one vertex. Since $v \in V(C)$ and C is connected, $N_G(v) \cap C \neq \emptyset$, which implies that $N_G(w) \cap C \neq \emptyset$ and consequently $w \in V(C)$. Moreover, since v and w are false twins in G, they are also false twins in C. This implies that the graph $C - v$ is connected and hence a component of $G' - S$. Note that D is an S-full component of $G' - S$. We complete the proof that S is a minimal separator in G' by showing that $C - v$ is also an S-full component of $G' - S$. Consider an arbitrary vertex $x \in S$. Since C is an S-full component of $G - S$, vertex x has a neighbor in C. However, since v and w are false twins in G, it cannot be that $N_G(x) \cap V(C) = \{v\}$. It follows that x also has a neighbor in $C - v$. Therefore, $C - v$ is an S-full component of $G' - S$, as claimed.

It remains to show that ϕ is one-to-one. Let S_1 and S_2 be distinct minimal separators in $\mathcal{S}_G^-$. If $S_1 \cap \{v\} \neq S_2 \cap \{v\}$, then $S_1 \cap \{w\} \neq S_2 \cap \{w\}$, inferring that $\phi(S_1) \neq \phi(S_2)$. If we have that $v \in S_1 \cap S_2$, then $\phi(S_1) = S_1 \setminus \{v\}$ and $\phi(S_2) = S_2 \setminus \{v\}$. Since $S_1 \neq S_2$ implies that $S_1 \setminus \{v\} \neq S_2 \setminus \{v\}$, we have that $\phi(S_1) \neq \phi(S_2)$. Finally, if none of the sets S_1, S_2 contain v, then ϕ is identity mapping, and $\phi(S_1) \neq \phi(S_2)$. It follows that ϕ is one-to-one, as claimed. This completes the proof. □

Next, we consider the deletion of a simplicial vertex. As noticed by Deogun et al. in [92, Lemma 13], no minimal separator in a graph G contains a simplicial vertex of G. In fact, the following bounds hold.

Proposition 3.2.15. *Let G be a graph and let v be a simplicial vertex in G. Then*

$$s(G - v) \leq s(G) \leq s(G - v) + 1\,.$$

Proof. Let $K = N_G(v)$ and $G' = G-v$. The inequality $s(G-v) \leq s(G)$ follows directly from Proposition 3.2.10. We will prove the second inequality by showing the inclusion $\mathcal{S}_G \subseteq \mathcal{S}_{G'} \cup \{K\}$.

Let $S \in \mathcal{S}_G$. If $S = K$, then $S \in \mathcal{S}_{G'} \cup \{K\}$, so we may assume that $S \neq K$. From Lemma 3.1.2 it follows that the graph $G - S$ has two S-full components C and D. Since $V(C)$ and $V(D)$ are anticomplete to each other in G and since $N_G(v)$ is a clique, it follows that $v \notin S$. Let C_v be the component of $G-S$ containing v. If $C_v \notin \{C, D\}$, then C and D are two S-full components of $G' - S$ and hence $S \in \mathcal{S}_{G'}$. We may thus assume that $C_v = C$. If $V(C) = \{v\}$, then $K \subseteq S$ and since C is an S-full component of $G-S$, it follows that $S = K$, a contradiction. So we have that $|V(C)| \geq 2$; moreover, since v is a simplicial vertex in a connected graph C, the graph $C-v$ is connected. Note that D is an S-full component of $G' - S$. We complete the proof that S is a minimal separator in G' by showing that $C - v$ is also an S-full component of $G' - S$. Suppose for a contradiction that this is not the case, that is, there exists a vertex $x \in S$ without a neighbor in $C - v$. Since C is an S-full component of $G - S$, vertex x has a neighbor in C. Hence, $N_G(x) \cap V(C) = \{v\}$. However, since C is a connected graph containing v, it follows that $N_G(v) \cap V(C) \neq \emptyset$. Taking an arbitrary $w \in N_G(v) \cap V(C)$, we now obtain that x and w are a pair of non-adjacent neighbors of v in G, contradicting the fact that v is a simplicial vertex in G. Therefore, $C-v$ is an S-full component of $G'-S$, as claimed, and hence S is a minimal separator in G'; in particular, $S \in \mathcal{S}_{G'} \cup \{K\}$. Since $S \in \mathcal{S}_G$ was arbitrary, this shows $\mathcal{S}_G \subseteq \mathcal{S}_{G'} \cup \{K\}$. $\square$

Remark 3.2.16. It is not difficult to construct examples showing that each of the inequalities in Proposition 3.2.15 can be attained with equality. For example, if $G = P_3$ and $v \in V(G)$ is of degree one, then $s(G - v) = 0$ and $s(G) = 1$, while if G is the paw and $v \in V(G)$ is of degree two, then $s(G - v) = s(G) = 1$.

Chapter 4

Tame graph classes

In this chapter we study graphs with "few" minimal separators. Recall that a graph class is said to be *tame* if graphs in the class have a polynomially bounded number of minimal separators. Since many problems that are NP-hard for general graphs become polynomial-time solvable for tame classes of graphs, the identification of tame graph classes is a natural research question.

Our results from this chapter can be summarized as follows:

(1) We analyze operations on graphs that preserve tame classes of graphs.

(2) We summarize known classes of graphs that are not tame and identify the conditions that should be fulfilled by every tame graph class.

(3) We characterize tame graph classes within the family of graph classes defined by sets of forbidden induced subgraphs with at most four vertices.

(4) We characterize tame graph classes within the family of graph classes defined by a single forbidden induced minor or a single forbidden induced topological minor.

Some results presented in this chapter are based on results from the following paper: [183] Milanič, M., Pivač, N. Polynomially Bounding the Number of Minimal Separators in Graphs: Reductions, Sufficient Conditions, and a Dichotomy Theorem. Electron. J. Combinatorics, 28(1):Paper No. 1.41, 27 (2021). https://doi.org/10.37236/9428

4.1 Introduction

The number of minimal separators in a graph on n vertices clearly depends on the structure of a graph. It can be as low as zero (if the graph is complete), but in general, it can be exponential in terms of the number of vertices of the graph. Consider, for example, a graph consisting of two non-adjacent vertices a and b and k internally disjoint paths of length three between them. Any vertex set consisting of an internal vertex from each of k paths is a minimal (a, b)-separator in such a graph, hence, the number of distinct minimal separators is at least 2^k.

It is thus an interesting question when this number can be bounded by a polynomial, and such graph classes enjoy good algorithmic properties.

Definition 4.1.1. *A graph class $\mathcal{G}$ is* tame *if there exists a polynomial $p : \mathbb{R} \to \mathbb{R}$ such that for every graph $G \in \mathcal{G}$, we have $s(G) \leq p(|V(G)|)$.*

A graph class is *feral* if there exists a constant $c > 1$ so that for arbitrarily large n there is an n-vertex graph in the class with at least c^n minimal separators. We begin this study with two straightforward observations about tame graph classes.

Observation 4.1.2. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two graph classes such that $\mathcal{G}_1 \subseteq \mathcal{G}_2$. If $\mathcal{G}_2$ is tame, then so is $\mathcal{G}_1$.*

Definition 4.1.3. *Given a non-negative integer k, we say that a graph class $\mathcal{G}$ is* k-tame *if $s(G) \leq |V(G)|^k - 1$ for every graph $G \in \mathcal{G}$.*

The reason for including the term '-1' in the definition of k-tame graph classes is purely technical, as it simplifies some of the statements and proofs (for example, those of Lemma 4.2.4).

Lemma 4.1.4. *A graph class $\mathcal{G}$ is tame if and only if it is k-tame for some non-negative integer k.*

Proof. Sufficiency is trivial. To prove necessity, let $\mathcal{G}$ be a tame graph class and let $p(x) = \sum_{i=0}^{d} a_i x^i$ be a polynomial such that $s(G) \leq p(|V(G)|)$ for all $G \in \mathcal{G}$. We may assume that $a_i \geq 0$ for all i, since otherwise we may delete the terms of p with negative coefficients to obtain a polynomial q such that $s(G) \leq q(|V(G)|)$ for all $G \in \mathcal{G}$. Moreover, we may assume that $a_0 = \ldots = a_d$, since otherwise, as long as there is a pair (i, j) with $0 \leq i < j \leq d$ and $a_i < a_j$, we may increase the i-th coefficient from a_i to a_j to obtain a polynomial q such that $s(G) \leq q(|V(G)|)$ for all $G \in \mathcal{G}$. Let a be this common value, that is, $a_0 = \ldots = a_d = a$. We thus have $p(x) = a(\sum_{i=0}^{d} x^i)$ and hence $p(n) \leq an^{d+1}$ holds for all $n \geq 2$. Let ℓ be the least non-negative integer such that $a \leq 2^\ell$. Then, for all $n \geq 2$, we have $an^{d+1} \leq 2^\ell \cdot n^{d+1} \leq n^\ell \cdot n^{d+1} = n^{d+\ell+1} \leq n^{d+\ell+2} - 1$. Since the 1-vertex graph has no minimal separators, it follows that all $G \in \mathcal{G}$ satisfy $s(G) \leq |V(G)|^{d+\ell+2} - 1$. Thus, taking $k = d + \ell + 2$, necessity is proved. $\square$

In the proofs of results from this chapter we will use some known results regarding tame graph classes. We put particular emphasis on the classes of P_4-free graphs and $2P_2$-free graphs. The following result is a consequence of more general results due to Nikolopoulos and Palios [187] and Pedrotti and de Mello [196].

Theorem 4.1.5. *The class of P_4-free graphs is tame.*

In Chapter 3 we saw that every minimal separator in a $2P_2$-free graph G is a neighborhood of some vertex in G. This in particular implies that a $2P_2$-free graph G can have at most $|V(G)|$ minimal separators. We state below this result for later use.

Corollary 4.1.6. *The class of $2P_2$-free graphs is tame.*

4.2 Graph operations and tame graph classes

In this section, we study operations on graph classes preserving tameness. We start with the relations between family of minimal separators of a disconnected graph and the families of minimal separators of its components. Note that some results presented in this section are corollaries of the results presented in Section 3.2.

Corollary 4.2.1. *Let $\mathcal{G}$ be a hereditary graph class and let $\mathcal{G}'$ be the class of connected graphs in $\mathcal{G}$. Then $\mathcal{G}$ is tame if and only if $\mathcal{G}'$ is tame.*

Proof. Since $\mathcal{G}$ is hereditary, we have $\mathcal{G}' \subseteq \mathcal{G}$. Hence, if $\mathcal{G}$ is tame, then so is $\mathcal{G}'$ by Observation 4.1.2. Suppose that $\mathcal{G}'$ is tame. By Lemma 4.1.4, there exists a positive integer k such that $s(G) \leq |V(G)|^k - 1$ for all $G \in \mathcal{G}'$. Let $G \in \mathcal{G} \setminus \mathcal{G}'$ and let $G_1, \ldots, G_p$ (with $p \geq 2$) be the components of G. Since G is disconnected and for all $i \in \{1, \ldots, p\}$ we have $G_i \in \mathcal{G}'$, we infer using Corollary 3.2.2 that $s(G) = \sum_{i=1}^p s(G_i) + 1 \leq \sum_{i=1}^p (|V(G_i)|^k - 1) + 1 \leq \sum_{i=1}^p |V(G_i)|^k - 1 \leq (\sum_{i=1}^p |V(G_i)|)^k - 1 = |V(G)|^k - 1$. It follows that $\mathcal{G}$ is tame. □

Note that the above proof also shows that for every positive integer k, the class $\mathcal{G}$ is k-tame if and only if $\mathcal{G}'$ is k-tame.

Corollary 4.2.2. *Let $\mathcal{G}$ be a hereditary graph class and let $\mathcal{G}^*$ be the class of coconnected graphs in $\mathcal{G}$. Then $\mathcal{G}$ is tame if and only if $\mathcal{G}^*$ is tame.*

Proof. Since $\mathcal{G}$ is hereditary, we have $\mathcal{G}^* \subseteq \mathcal{G}$. Thus, if $\mathcal{G}$ is tame, then so is $\mathcal{G}^*$ by Observation 4.1.2. Suppose that $\mathcal{G}^*$ is tame. By Lemma 4.1.4, there exists a positive integer k such that $s(G) \leq |V(G)|^k - 1$ for all $G \in \mathcal{G}^*$. Let $G \in \mathcal{G} \setminus \mathcal{G}^*$ and let $G_1, \ldots, G_p$ (with $p \geq 2$) be the cocomponents of G. Since for all $i \in \{1, \ldots, p\}$ we have $G_i \in \mathcal{G}^*$, we infer using Corollary 3.2.4 and the assumption on $\mathcal{G}^*$ that $s(G) = \sum_{i=1}^p s(G_i) \leq \sum_{i=1}^p (|V(G_i)|^k - 1) \leq \sum_{i=1}^p |V(G_i)|^k - 1 \leq (\sum_{i=1}^p |V(G_i)|)^k - 1 = |V(G)|^k - 1$. It follows that $\mathcal{G}$ is tame. □

Note that the above proof also shows that for every positive integer k, the class $\mathcal{G}$ is k-tame if and only if $\mathcal{G}^*$ is k-tame. Recall that a graph is said to be an *atom* if it has no clique cutset and that, given a class $\mathcal{G}$ of graphs, we denote by $A(\mathcal{G})$ the class of all atoms that are induced subgraphs of a graph in $\mathcal{G}$. Thus, Theorem 4.2.3 is a direct consequence of Lemma 4.1.4.

Theorem 4.2.3. *For every graph class $\mathcal{G}$, if $A(\mathcal{G})$ is tame, then so is $\mathcal{G}$.*

Lemma 4.2.4. *Let $\mathcal{G}$ be a graph class such that the class of atoms $A(\mathcal{G})$ is k-tame for some non-negative integer k. Then $\mathcal{G}$ is $(k+1)$-tame.*

Proof. Let $\mathcal{G}$ and k be as in the lemma. We show that every n-vertex graph $G \in \mathcal{G}$ has at most $n^{k+1} - 1$ minimal separators. The proof is by induction on n.

If $n = 1$, then $G = K_1$ and G has $0 = 1^{k+1} - 1$ minimal separators. Suppose that $n > 1$ and let G be an n-vertex graph from $\mathcal{G}$. If G is an atom, then G has at most $n^k - 1$ minimal separators by assumption, and $n^k - 1 < n^{k+1} - 1$. Suppose now that G has a clique cutset. Then, G has a cut-partition (A, B, C) such that C is a clique and $G_A = G[A \cup C]$ has no clique cutset (see, e.g., [36, 226]). Since G_A belongs to $A(\mathcal{G})$, we have $s(G_A) \leq |V(G_A)|^k - 1$. Furthermore, since B is non-empty, we have $|V(G_A)| \leq n-1$ and consequently, $s(G_A) \leq (n-1)^k - 1$. Note that also A is non-empty, and hence we can apply the induction hypothesis to the graph $G_B = G[B \cup C] \in \mathcal{G}$ to derive $s(G_B) \leq |V(G_B)|^{k+1} - 1 \leq (n-1)^{k+1} - 1$. By Lemma 3.2.5, we have

$$s(G) \leq s(G_A) + s(G_B) + 1 \leq (n-1)^k + (n-1)^{k+1} - 1\,.$$

Thus, to complete the proof, it suffices to show the following inequality:

$$n^{k+1} - (n-1)^{k+1} \geq (n-1)^k \,. \tag{4.1}$$

Note that for every two non-negative real numbers a and b we have

$$a^{k+1} - b^{k+1} = (a-b) \cdot \left(\sum_{i=0}^{k} a^{k-i} b^i \right) \geq (a-b) b^k \,.$$

Applying the inequality $a^{k+1} - b^{k+1} \geq (a-b)b^k$ to $a = n$, $b = n-1$ establishes (4.1). □

4.3 Some non-tame graph classes

In this section we give an overview of known constructions of graphs with exponentially many minimal separators. Obviously, such constructions imply certain necessary conditions for a graph class to be tame, in the sense that graphs in a tame graph class cannot contain the corresponding structures with respect to some graph containment relation. First we will describe two families of graphs with exponentially many minimal separators that were presented in [183]. In particular, the first construction involves families of graphs of arbitrarily large maximum degree but without long induced paths. The second construction involves two families of graphs with small maximum degree but with arbitrarily long induced paths. In both cases, we make use of line graphs.

Given positive integers k and ℓ, the *k,ℓ-theta graph* is the graph $\theta_{k,\ell}$ obtained as the union of k internally disjoint paths of length ℓ with common endpoints a and b. For every positive integer ℓ, we define a family of graphs Θ_ℓ in the following way: $\Theta_\ell = \{\theta_{k,\ell} \mid k \geq 2\}$. Note that ℓ refers to the length of each of the a,b-paths and not to the number of paths, which is unrestricted.

Observation 4.3.1. *For every integer $\ell \geq 3$, the class Θ_ℓ is not tame.*

Proof. Let $k \geq 2$, $\ell \geq 3$, let $G = \theta_{k,\ell}$, and let $P^1, \ldots, P^k$ be paths in G as in the definition of the theta graphs. Let S be any set of vertices of G containing exactly one internal vertex of each of the paths P^j. Then, the graph $G - S$ has two S-full components and Lemma 3.1.2 implies that S is a minimal separator in G. Note that for every $j \in \{1, \ldots, k\}$, path P^j has exactly $\ell - 1$ internal vertices. It follows that $s(\theta_{k,\ell}) \geq (\ell-1)^k$. Thus, as $|V(\theta_{k,\ell})| = k(\ell-1)+2$, we infer that for every fixed positive integer $\ell \geq 3$, the class Θ_ℓ is not tame. □

Corollary 4.3.2. *If $\mathcal{G}$ is graph class such that $\Theta_\ell \subseteq \mathcal{G}$ for some $\ell \geq 3$, then $\mathcal{G}$ is not tame.*

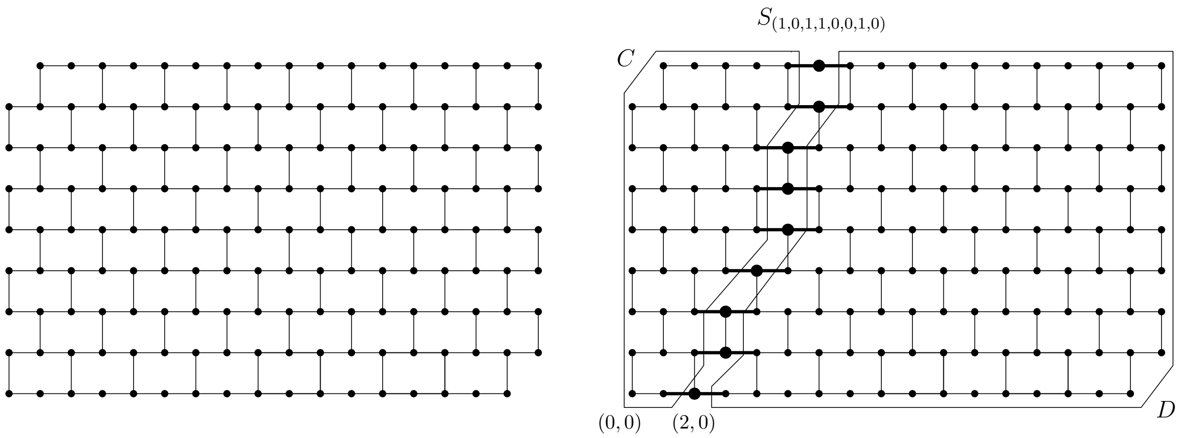
Consider now the family of line graphs of theta graphs. More precisely, given positive integers k and ℓ, let $L_{k,\ell}$ denote the line graph of $\theta_{k,\ell}$ and let $\mathcal{L}_\ell = \{L_{k,\ell} \mid k \geq 2\}$. Note that the class $\mathcal{L}_2$ is precisely the class of all short prisms of order at least two.

Observation 4.3.3. *For every integer $\ell \geq 2$, the class $\mathcal{L}_\ell$ is not tame.*

Proof. Let $k, \ell \geq 2$ and let $G = L_{k,\ell}$. Then, graph G consists of two cliques K and K', each of size k, say $K = \{a_1, \ldots, a_k\}$ and $K' = \{b_1, \ldots, b_k\}$, and k internally pairwise disjoint paths $P^1, \ldots, P^k$ such that for every $j \in \{1, \ldots, k\}$, path P^j is an a_i, b_i-path with $|V(P^j)| = \ell$, $V(P^j) \cap K = \{a_j\}$ and $V(P^j) \cap K' = \{b_j\}$. Consider any set S of vertices of G containing exactly one vertex from each of the paths P^j and such that $S \notin \{K, K'\}$. Then, the graph $G - S$ has two S-full components and Lemma 3.1.2 implies that S is a minimal separator in G. It follows that $s(L_{k,\ell}) \geq \ell^k - 2$. Thus, as $|V(L_{k,\ell})| = k(\ell + 1)$, we infer that for every fixed positive integer $\ell \geq 2$, the class $\mathcal{L}_\ell$ is not tame. $\square$

Corollary 4.3.4. *If $\mathcal{G}$ is a graph class such that $\mathcal{L}_\ell \subseteq \mathcal{G}$ for some $\ell \geq 2$, then $\mathcal{G}$ is not tame.*

We now turn to the second type of construction for families of graphs with exponentially many minimal separators. Let $r, s \geq 2$ be integers. An *$r \times s$-grid* is the graph with vertex set $\{0, \ldots, r-1\} \times \{0, \ldots, s-1\}$ in which two vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| + |j - j'| = 1$. Given an integer $h \geq 2$, an *elementary wall of height h* is the graph W_h obtained from the $(2h+2) \times (h+1)$-grid by deleting all edges with endpoints $(2i+1, 2j)$ and $(2i+1, 2j+1)$ for all $i \in \{0, 1, \ldots, h\}$ and $j \in \{0, 1, \ldots, \lfloor (h-1)/2 \rfloor\}$, deleting all edges with endpoints $(2i, 2j-1)$ and $(2i, 2j)$ for all $i \in \{0, 1, \ldots, h\}$ and $j \in \{1, \ldots, \lfloor h/2 \rfloor\}$, and deleting the two resulting vertices of degree one. Note that an elementary wall of height h consists of h levels each containing h bricks, where a brick is a cycle of length six; see Fig. 4.1(a).



(a) (b)

Figure 4.1: (a) An elementary wall of height 8. (b) A minimal separator $S_{(1,0,1,1,0,0,1,0)}$ in W_8 and the two components of $W_8 - S_{(1,0,1,1,0,0,1,0)}$.

Grids contain exponentially many minimal separators [221]. We show next that the same is true for walls.

Proposition 4.3.5. *For every integer $h \geq 2$, an elementary wall of height h has at least 2^h minimal separators.*

Proof. Fix an integer $h \geq 2$. We will define a family of 2^h subsets of $V(W_h)$ and show that each of them is a minimal separator in W_h. For each binary sequence of length h, say $x = (x_1, \ldots, x_h) \in \{0,1\}^h$, we define a set S_x by the following rule: $S_x = \{v^{x,0}, v^{x,1}, \ldots, v^{x,h}\}$ where $v^{x,0} = (2,0)$ (independently of x) and for all $j \in \{1,\ldots,h\}$, we set $v^{x,j} = v^{x,j-1} + (x_j, 1)$, where addition is performed component-wise. Clearly, for each $x \in \{0,1\}^h$ and each $j \in \{1,\ldots,h\}$, we have $v^{x,j} = (\sum_{i=1}^{j} x_i + 2, j) \leq (h+2, h)$, where comparison is performed component-wise. It follows that $S_x \subseteq V(W_h)$. Moreover, the graph $W_h - S_x$ has exactly two connected components, say C and D, with $V(C) = \bigcup_{j=0}^{h}\{(i,j) \in V(W_h) \mid i < v_1^{x,j}\}$ and $V(D) = \bigcup_{j=0}^{h}\{(i,j) \in V(W_h) \mid i > v_1^{x,j}\}$. Note that each vertex $v^{x,j} \in S_x$ has a neighbor in C, namely $v^{x,j} - (1,0)$, and a neighbor in D, namely $v^{x,j} + (1,0)$. By Lemma 3.1.2, set S_x is a minimal separator in W_h. Since the sets S_x are pairwise distinct, this completes the proof. Fig. 4.1(b) shows an example with $h = 4$ and $x = (1,0,1,1)$. The thick horizontal edges can be used to justify the fact that C and D are S_x-full components of $W_h - S_x$. $\square$

Another family with exponentially many minimal separators is given by the line graphs of elementary walls; see Fig. 4.2(a) for an example.

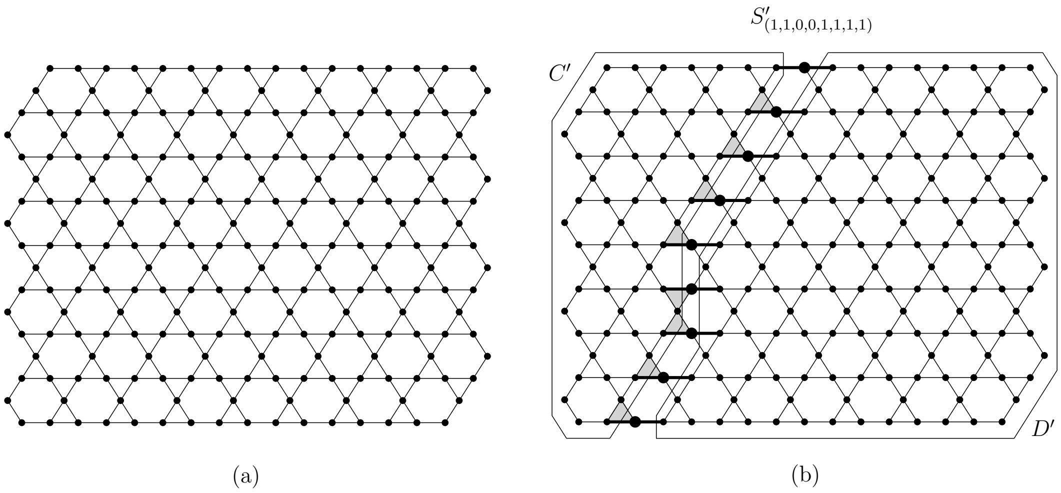


Figure 4.2: (a) $L(W_8)$, the line graph of an elementary wall of height 8. (b) The set of nine vertices depicted with large black disks is a minimal separator $S'_{(1,1,0,0,1,1,1,1)}$ in $L(W_8)$, which corresponds to the minimal separator $S_{(1,1,0,0,1,1,1,1)}$ in W_8. The two components of $L(W_8) - S'_{(1,1,0,0,1,1,1,1)}$ are also depicted.

Proposition 4.3.6. *For every even integer $h \geq 2$, the graph $L(W_h)$ has at least $2^{h/2}$ minimal separators.*

Proof. We use a modification of the construction used in the proof of Proposition 4.3.5. We again consider the minimal separators S_x in W_h constructed in the proof of Proposition 4.3.5; however, for technical reasons that will simplify the argument, we restrict ourselves only to the $2^{h/2}$ minimal separators S_x in W_h that arise from binary sequences $x \in X_h$, where

$$X_h = \{(x_1, \ldots, x_h) \in \{0,1\}^h \mid x_{2i-1} = x_{2i} \text{ for all } i \in \{1,2,\ldots,h/2\}\}.$$

Recall that for every $x \in X_h$, we have $S_x = \{v^{x,0}, v^{x,1}, \ldots, v^{x,h}\}$ where $v^{x,0} = (2,0)$ and $v^{x,j} = v^{x,j-1} + (x_j, 1)$ for all $j \in \{1,\ldots,h\}$. A set of $2^{h/2}$ minimal separators of $L(W_h)$

can be obtained as follows. For each $x \in X_h$, we define a set $S'_x \subseteq V(L(W_h))$ as follows: $S'_x = \{e^{x,j} \mid v^{x,j} \in S_x\}$ where $e^{x,j}$ is the vertex of the line graph of W_h corresponding to the edge in W_h joining vertex $v^{x,j}$ with vertex $v^{x,j} + (1,0)$.

Since the mapping is one-to-one, the set $\{S'_x \mid x \in X_h\}$ is of cardinality $2^{h/2}$. Therefore, to complete the proof it suffices to show that for every $x \in X_h$, set S'_x is a minimal separator in $L(W_h)$. Let us first argue that the graph $L(W_h) - S'_x$ is disconnected. Vertices of the wall W_h correspond bijectively to maximal cliques of its line graph. For every $x \in X_h$, every vertex of the form $v^{x,j}$ where $j \in \{1, \ldots, h-1\}$ corresponds to a triangle (clique of size three) in $L(W_h)$, while vertex $v^{x,h}$ corresponds to a clique of size two. Let us say that a triangle in $L(W_h)$ is *upward pointing* if it arises from a vertex in W_h whose coordinates have even sum, and *downward pointing*, otherwise. (We draw this terminology from the planar embeddings of the line graphs of the walls following the example given in Fig. 4.2.) It is not difficult to see that for every $x \in X_h$ and every even $i \in \{0, 1, \ldots, h-2\}$, vertex $v^{x,i}$ corresponds to an upward triangle, while odd-indexed vertices may correspond to either upward or downward pointing triangles. It follows that for no index $i \in \{0, 1, \ldots, h-1\}$, vertices $v^{x,i}$ and $v^{x,i+1}$ can both correspond to downward pointing triangles. This property ensures that the graph $L(W_h) - S'_x$ is disconnected, with exactly two components C' and D' such that for all $v^{x,j} \in S_x$, component C' contains all vertices of the form e^{x,j^-}, where $e^{x,j^-} \in V(L(W_h))$ is the vertex corresponding to the edge in W_h joining vertex $v^{x,j}$ with vertex $v^{x,j} - (1,0)$, while component D' contains all vertices of the form e^{x,j^+}, where $e^{x,j^+} \in V(L(W_h))$ is the vertex corresponding to the edge in W_h joining vertex $v^{x,j} + (1,0)$ with vertex $v^{x,j} + (2,0)$. Furthermore, since for every vertex $e^{x,j} \in S'_x$, vertices e^{x,j^-} and e^{x,j^+} are both adjacent to $e^{x,j}$ in $L(W_h)$, this also implies, by Lemma 3.1.2, that S'_x is a minimal separator in $L(W_h)$. This completes the proof. Fig. 4.2(b) shows an example with $h = 8$ and $x = (1,1,0,0,1,1,1,1)$. The thick horizontal edges can be used to justify the fact that C' and D' are S'_x-full components of $L(W_h) - S'_x$. □

The above proofs were originally presented by Milanič and Pivač in [183,184]. Their initial systematic study of tame graph classes was followed by a series of other results. Using the subsequent recent literature, we perform a brief survey of graph classes that lack the property of tameness, and fix some terminology (see also Chapter 2). Abrishami et al. [3] obtained new results concerning tame graph classes and provided some examples of graph families having exponentially many minimal separators. Some of the families listed in their work were already presented in an earlier paper by Chudnovsky et al. [66].

In particular, Abrishami et al. considered graphs called *k-theta* and *k-prism*, as follows. A *k-theta* is a graph obtained as the union of k internally disjoint paths of length 3 with common endpoints a and b. More precisely, a k-theta is a graph G with vertex set $V(G) = \{a, a_1, \ldots, a_k, b, b_1, \ldots, b_k\}$, and edge set consisting of the pairs of the following form: aa_i, bb_i, and a_ib_i for $1 \leq i \leq k$ (see Fig. 4.3; these graphs were depicted in Fig. 2.1 as well, although we reproduce some part of it for convenience). Any graph that is a k-theta for some $k \geq 3$ will be referred to as a *short theta*. A *k-prism* is a graph whose vertex set can be partitioned into two n-vertex cliques, say $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$, such that for all $i, j \in \{1, \ldots, n\}$, a_i is adjacent to b_j if and only if $i = j$. Any graph that is a k-prism for some $k \geq 3$ will be referred to as *short prism*. Note that the class consisting of all short thetas is precisely the class

Θ_3 and the class of all short prisms is precisely the class $\mathcal{L}_2$.

The class of all short prisms and the class of all short thetas are known to be feral, which implies they are not tame (see, e.g., [3, 183]). Combining the structure of k-thetas and k-prisms results in a graph called a k*-pyramid*, that is, a graph G with vertex set $V(G) = \{a, a_1, \ldots, a_k, b_1, \ldots, b_k\}$, and with the edge set consisting of the pairs of the following form: aa_i and a_ib_i for $1 \leq i \leq k$, and b_ib_j for $1 \leq i < j \leq k$. It is not difficult to see that a k-pyramid has at least 2^{k-1} minimal separators, with similar argumentation as described for k-thetas and k-prisms.

To avoid terminological conflict, let us recall that a *theta* is any subdivision of the complete bipartite graph $K_{2,3}$ and *prism* is any subdivision of $\overline{C_6}$ in which the two triangles remain unsubdivided, as defined in Chapter 2. A *pyramid* is any subdivision of the complete graph K_4 in which one triangle remains unsubdivided, and of the remaining three edges, at least two edges are subdivided at least once.

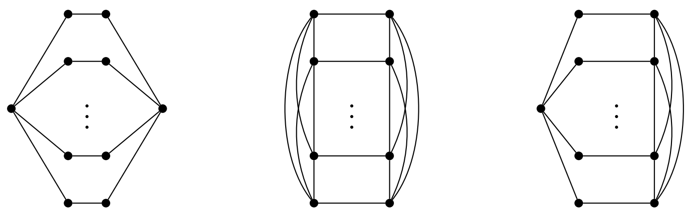


Figure 4.3: k-theta, k-prism, k-pyramid.

A k*-turtle* is a graph G with two non-adjacent vertices $a, b \in V(G)$, two paths P and Q from a to b, vertex-disjoint except for a and b, such that $V(P) \cup V(Q)$ induces a cycle H in G. Also, for $1 \leq i \leq k$, $x_i, y_i \in V(G) \setminus V(H)$ such that $x_iy_i \in E(G)$, and x_i has at least three neighbors in P and no neighbors in Q, and y_i has at least three neighbors in Q and no neighbors in P. Chudnovsky et al. [66] provided a construction of k-turtles as the example of a graph family that is (theta, prism, pyramid)-free and contains exponentially many minimal separators. Every k-turtle contains a smaller graph, called turtle, defined as follows (see Fig. 4.4). A *turtle* is a graph G consisting of two vertex-disjoint paths P and Q and two adjacent vertices $x, y \in V(G) \setminus (V(P) \cup V(Q))$ such that P is a path from a_1 to b_1, Q is a path from a_2 to b_2, a_1 is adjacent to a_2, b_1 is adjacent to b_2, $V(P) \cup V(Q)$ induces a hole in G, x has at least three neighbors in P and no neighbors in Q, and y has at least three neighbors in Q and no neighbors in P.

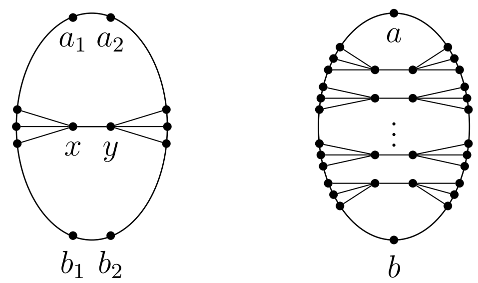


Figure 4.4: The turtle and k-turtle.

In [66], the authors conjectured that forbiding the aforementioned constructions represents sufficient conditions for a graph class to be tame. Soon after, Abrishami et al. proved this conjecture and showed that there is a polynomial p such that every graph G that contains no prism, pyramid, theta, or turtle has at most $p(|V(G)|)$ minimal separators [3].

In the same work (see [3]), the authors introduced a structure called k-creature, where a *k-creature* in a graph G is a tuple (A, B, X, Y) of pairwise disjoint nonempty vertex sets such that (i) A and B induce connected subgraphs, (ii) A is anticomplete to $Y \cup B$ and B is anticomplete to $A \cup X$, (iii) every $x \in X$ has a neighbor in A and every $y \in Y$ has a neighbor in B, (iv) $|X| = |Y| = k$ and X and Y can be enumerated as $X = \{x_1, \ldots, x_k\}$, $Y = \{y_1, \ldots, y_k\}$ so that $x_i y_i \in E(G)$ if and only if $i = j$ (see Fig. 4.5). We say that G is k-creature-free if G does not contain a k-creature as an induced subgraph. Note that this structure generalizes the structure of k-thetas, k-prisms ($k \geq 3$), and k-pyramids, so it is easy to verify that a k-creature contains at least 2^k minimal separators.

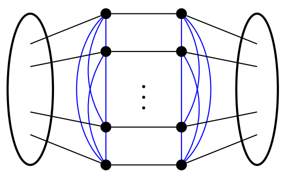


Figure 4.5: k-creature. Blue edges may or may not exist.

The structures presented in this section turn out to be crucial for showing the boundaries between tame and non-tame graph classes, as we will see in the following sections.

4.4 Characterization of tame graph classes with small forbidden induced subgraphs

In this section we completely characterize which graph classes defined by forbidden induced subgraphs with at most four vertices are tame. To describe the result, we need to introduce some notation. Given two families $\mathcal{F}$ and $\mathcal{F}'$ of graphs, we write $\mathcal{F} \trianglelefteq \mathcal{F}'$ if the class of $\mathcal{F}$-free graphs is contained in the class of $\mathcal{F}'$-free graphs, or, equivalently, if every $\mathcal{F}$-free graph is also $\mathcal{F}'$-free. It is well known and not difficult to see that the relation $\mathcal{F} \trianglelefteq \mathcal{F}'$ can be checked by means of the following criterion, which becomes particularly simple for finite families $\mathcal{F}$ and $\mathcal{F}'$.

Observation 4.4.1. *For every two graph families $\mathcal{F}$ and $\mathcal{F}'$, we have $\mathcal{F} \trianglelefteq \mathcal{F}'$ if and only if every graph from $\mathcal{F}'$ contains an induced subgraph isomorphic to a member of $\mathcal{F}$.*

In order to obtain a dichotomy result that characterizes sets of forbidden induced subgraphs on at most four vertices for tame graph classes, first we give some necessary conditions. In previous sections we proposed some graph families that have exponentially many minimal separators, and here we use them to derive some properties of a finite set of graphs $\mathcal{F}$ such that the class of $\mathcal{F}$-free graphs is tame. In particular, Corollaries 4.3.2 and 4.3.4 imply that any graph class $\mathcal{G}$ that contains the class Θ_ℓ for $\ell \geq 3$ or the class $\mathcal{L}_\ell$ for $\ell \geq 2$ is not tame.

Proposition 4.4.2. *Let $\mathcal{F}$ be a finite set of graphs such that the class of $\mathcal{F}$-free graphs is tame. Then $\mathcal{F}$ either contains an induced subgraph of P_4 or of $2P_2$, or $\mathcal{F}$ contains both an acyclic graph and a graph of girth at most 4.*

Proof. Let $\mathcal{F}$ be a finite set of graphs such that for every $F \in \mathcal{F}$ we have $F \not\subseteq_i P_4$, $F \not\subseteq_i 2P_2$. Suppose for a contradiction that either all graphs in $\mathcal{F}$ contain cycles or all of them are of girth more than 5. We analyze the two cases separately.

Case 1: all graphs in $\mathcal{F}$ contain cycles. Let ℓ be the smallest integer such that $\ell \geq 3$ and for every graph $F \in \mathcal{F}$, it holds that F does not contain an induced cycle of length exactly 2ℓ. Note that ℓ is well defined since $\mathcal{F}$ is finite. We claim that every graph in Θ_ℓ is $\mathcal{F}$-free. Suppose for a contradiction that for some $k \geq 2$, the graph $\theta_{k,\ell}$ contains an induced subgraph isomorphic to some $F \in \mathcal{F}$. Since F contains an induced cycle and every induced cycle contained in $\theta_{k,\ell}$ is of length 2ℓ, we infer that F contains an induced cycle of length 2ℓ. However, this contradicts the definition of ℓ. Thus, every graph in Θ_ℓ is $\mathcal{F}$-free, as claimed. By Corollary 4.3.2, the class of $\mathcal{F}$-free graphs is not tame, a contradiction.

Case 2: every graph in $\mathcal{F}$ is of girth more than five. We will show that in this case, every graph in $\mathcal{L}_2$ is $\mathcal{F}$-free. By Corollary 4.3.4 this will imply that the class of $\mathcal{F}$-free graphs is not tame, a contradiction. From the definition of $\mathcal{L}_2$ it follows that every graph in $\mathcal{L}_2$ has independence number two. Thus, to show that every graph in $\mathcal{L}_2$ is $\mathcal{F}$-free, it suffices to prove that $\alpha(F) \geq 3$ for all $F \in \mathcal{F}$. Suppose for a contradiction that $\alpha(F) \leq 2$ for some $F \in \mathcal{F}$. Then F is acyclic, since otherwise a shortest cycle in F would be of length at least 6, which would imply $\alpha(F) \geq 3$. Moreover, F has at most two connected components. If F is connected, then F is a tree with $\alpha(F) \leq 2$. In particular, the maximum degree of F is at most 2, hence F a path with at most four vertices, which implies $F \subseteq_i P_4$, a contradiction. If F has exactly two components, then the condition $\alpha(F) \leq 2$ implies that each component of F is a complete graph. However, since F is acyclic, each component of F is an induced subgraph of P_2. It follows that $F \subseteq_i 2P_2$, a contradiction. □

In the following section we prove a number of propositions, each giving a sufficient condition for a family $\mathcal{F}$ of graphs on at most 4 vertices such that the class of $\mathcal{F}$-free graphs is tame. Then, we use these results, along with constructions of graphs with exponentially many minimal separators, and obtain the main result in Section 4.4.2.

4.4.1 Some tame graph classes

Now we are able to identify several sufficient conditions for a graph class to be tame. The first two reveal two infinite families of tame graph classes, each parameterized by two positive integers.

Graphs in which all edges are almost dominating

Some of our proofs will make use of the following classical result due to Ramsey [209].

Ramsey's Theorem. *For every two positive integers k and ℓ, there exists a least positive integer $R(k, \ell)$ such that every graph with at least $R(k, \ell)$ vertices contains either a clique of size k or an independent set of size ℓ.*

Our first sufficient condition is an easy consequence of Ramsey's theorem.

Theorem 4.4.3. *For every two positive integers k and ℓ, the class of $\{P_2+kP_1, K_\ell+P_2\}$-free graphs is tame.*

We prove the theorem using an auxiliary lemma. For a non-negative integer k, we denote by $\mathcal{C}_k$ the class of graphs G such that for every edge $uv \in E(G)$, at most k vertices of G are adjacent to neither u nor v.

Lemma 4.4.4. *For every positive integer k, the class $\mathcal{C}_k$ is tame.*

Proof. Since $\mathcal{C}_i \subseteq \mathcal{C}_{i+1}$ for all $i \geq 0$, we may assume that $k \geq 1$. We will prove that for every minimal separator S in G, there exists a set $X \subseteq V(G)$ such that $|X| \leq k$ and $S = N_G(X)$. Clearly, this will imply that G has at most $\binom{|V(G)|}{k}$ minimal separators. Let S be a minimal separator in G and let C and D be two S-full components of $G-S$. Since $N_G(V(C)) = N_G(V(D)) = S$, it suffices to show that $|V(C)| \leq k$ or $|V(D)| \leq k$. Suppose that this is not the case. Then $|V(C)| \geq k+1$ and $|V(D)| \geq k+1$. Since $|V(C)| \geq k+1 \geq 2$ and C is connected, there is an edge $uv \in E(C)$. But then, G contains at least $|V(D)| \geq k+1$ vertices that are adjacent to neither u nor v, contrary to the fact that $G \in \mathcal{C}_k$. □

Proof of Theorem 4.4.3. Let G be a $\{P_2+kP_1, K_\ell+P_2\}$-free graph and let $r = R(\ell, k)$. By Lemma 4.4.4, it suffices to show that $G \in \mathcal{C}_{r-1}$. Suppose this is not the case. Then, G has an edge uv such that there exists a set X of r vertices of G such that every vertex in X is adjacent to neither u nor v. By Ramsey's theorem, there exists a set $Z \subseteq X$ such that Z is either a clique of size ℓ or an independent set of size k in G. But then the set $\{u, v\} \cup Z$ induces either a $K_\ell + P_2$ or $P_2 + kP_1$, respectively. Both cases lead to a contradiction. □

Graphs of bounded clique cover number excluding some short prism

We now prove that every hereditary class of graphs of bounded clique cover number that does not contain all short prisms is tame. We formulate it in an equivalent way that will facilitate our inductive proof. We denote by L_k the k-prism. For every two positive integers k and ℓ, let $\mathcal{C}_{k,\ell}$ denote the class of all L_k-free graphs with clique cover number at most ℓ.

Theorem 4.4.5. *For every two positive integers k and ℓ, the class $\mathcal{C}_{k,\ell}$ is tame.*

We prove Theorem 4.4.5 by induction on ℓ, with cases $\ell \in \{1, 2\}$ as the base cases. In the proof for the case $\ell = 2$, we make use of the following result, discovered independently by Alekseev [6], Balas-Yu [13], and Prisner [205].

Theorem 4.4.6. *For every positive integer k, every $\overline{kP_2}$-free graph G has $O(|V(G)|^{2k-2})$ maximal cliques.*

Lemma 4.4.7. *For every positive integer k, the class $\mathcal{C}_{k,2}$ is tame.*

Proof. Let G be an L_k-free graph with clique cover number at most 2 and let $\{A_1, A_2\}$ be a clique cover of G. We associate to G a graph G' obtained by swapping the roles of edges and non-edges between cliques A_1 and A_2. Formally, G' is defined as follows: $V(G') = V(G)$ and $E(G') = E_1 \cup E_2 \cup E_3$ where $E_1 = \{uv \mid u, v \in A_1, u \neq v\}$, $E_2 = \{uv \mid u, v \in A_2, u \neq v\}$, and $E_3 = \{uv \mid u \in A_1, v \in A_2, uv \notin E(G)\}$.

We prove that G has a polynomially bounded number of minimal separators in two steps. First we prove that if a set $S \subseteq V(G)$ is a minimal separator in G, then its complement $\overline{S} = V(G) \setminus S$ is a maximal clique in G'. Then, we show that G' is $\overline{kP_2}$-free. Finally, we invoke the result of Theorem 4.4.6 to infer that G' has $O(|V(G')|^{2k-2}) = O(|V(G)|^{2k-2})$ maximal cliques. Since the mapping $S \mapsto \overline{S}$ is one-to-one, this will imply that G has $O(|V(G)|^{2k-2})$ minimal separators.

Let S be a minimal separator in G. Then $G - S$ has precisely two components, namely $G[A_1 \setminus S]$ and $G[A_2 \setminus S]$, and both of these components are S-full. Since sets $A_1 \setminus S$ and $A_2 \setminus S$ are anticomplete to each other in G, we infer that $\overline{S} = (A_1 \setminus S) \cup (A_2 \setminus S)$ is a clique in G'. It remains to prove that $\overline{S}$ is a maximal clique. Assume for a contradiction that there exists a vertex $x \in S$ such that $\overline{S} \cup \{x\}$ is a clique in G'. By symmetry we may assume that $x \in A_1 \cap S$. Since $A_2 \setminus S$ is an S-full component in $G - S$, vertex x is adjacent in G to some vertex $y \in A_2 \setminus S$. This implies that vertices x and y are non-adjacent in G', contradicting the fact that they both belong to clique $\overline{S} \cup \{x\}$. It follows that $\overline{S}$ is a maximal clique in G', as claimed.

To complete the proof of the lemma, it remains to show that G' is $\overline{kP_2}$-free. Assume the opposite: let $X \subseteq A_1, Y \subseteq A_2$ be such that $G'[X \cup Y] \cong \overline{kP_2}$. Since X and Y are cliques in G', all the non-edges of the $\overline{kP_2}$ must go from X to Y. It follows that $G[X \cup Y] \cong L_k$, contradicting the fact that G is L_k-free. Hence, G' is $\overline{kP_2}$-free, as claimed. □

Before proceeding to the induction step, we prove two more technical results. For two positive integers k, ℓ, let us denote by $\mathcal{C}^*_{k,\ell}$ the class of all graphs of the form $G - S^*$ such that $G \in \mathcal{C}_{k,\ell}$ and there exists a clique cover $\{A_1, \ldots, A_\ell\}$ of G such that S^* is a minimal separator in the graph $G - A_\ell$.

Lemma 4.4.8. *Suppose that for some positive integers k, ℓ with $\ell \geq 3$, the classes $\mathcal{C}_{k,\ell-1}$ and $\mathcal{C}^*_{k,\ell}$ are both tame. Then, the class $\mathcal{C}_{k,\ell}$ is also tame.*

Proof. Fix positive integers k, ℓ with $\ell \geq 3$ and suppose that the classes $\mathcal{C}_{k,\ell-1}$ and $\mathcal{C}^*_{k,\ell}$ are a-tame and b-tame, respectively. We want to prove that $\mathcal{C}_{k,\ell}$ is tame as well. Let G be a graph in $\mathcal{C}_{k,\ell}$ and let $\{A_1, A_2, \ldots, A_\ell\}$ be a clique cover of G. Let S be a minimal separator in G and let C and D be two distinct S-full components in $G - S$. Since $\ell \geq 3$, there exists at least one clique A_i, $i \in \{1, \ldots, \ell\}$, such that $V(C) \nsubseteq A_i$ and $V(D) \nsubseteq A_i$. By renumbering the cliques, if necessary, we may assume that $V(C) \nsubseteq A_\ell$ and $V(D) \nsubseteq A_\ell$. Furthermore, since C and D are anticomplete to each other and A_ℓ is a clique, we have $V(C) \cap A_\ell = \emptyset$ or $V(D) \cap A_\ell = \emptyset$. By symmetry, we may assume that $V(C) \cap A_\ell = \emptyset$.

Consider the graph $G - A_\ell$. Fix two vertices $u \in V(C)$ and $v \in V(D) \setminus A_\ell$. From the definition of S it follows that $S \setminus A_\ell$ is a (u, v)-separator in $G - A_\ell$. Let $S^* \subseteq S \setminus A_\ell$ be a minimal (u, v)-separator in $G - A_\ell$. By Corollary 3.2.9 it follows that the set $S' := S \setminus S^*$ is a minimal separator in $G - S^*$. In particular, we have $S = S^* \cup S'$ where $S^* \in \mathcal{S}_{G - A_\ell}$ and $S' \in \mathcal{S}_{G - S^*}$. Clearly, $G - A_\ell \in \mathcal{C}_{k,\ell-1}$. It follows that every minimal separator S in G can be written as a union of two sets S^* and S' such that $S^* \in \mathcal{S}_{G^*}$ where $G^* = G - A_\ell \in \mathcal{C}_{k,\ell-1}$ and $S' \in \mathcal{S}_{G'}$ where $G' = G - S^* \in \mathcal{C}^*_{k,\ell}$.

Note that the above arguments hold for any clique cover $\{A_1, A_2, \ldots, A_\ell\}$ of G, except that some renaming of cliques might have been necessary depending on S, C, and D, to assure that $V(C) \cap A_\ell = \emptyset$ and $V(D) \nsubseteq A_\ell$. Once a clique cover $\{A_1, A_2, \ldots, A_\ell\}$ of G is fixed, there are ℓ choices for which a clique in the cover is labeled

A_ℓ. Furthermore, since $\mathcal{C}_{k,\ell-1}$ is a-tame, there are at most $|V(G^*)|^a - 1 \leq |V(G)|^a - 1$ choices for S^*. The graph $G' \in \mathcal{C}^*_{k,\ell}$ as above is uniquely determined with G and S^* and since $\mathcal{C}^*_{k,\ell}$ is b-tame, there are at most $|V(G')|^b - 1 \leq |V(G)|^b - 1$ choices for S'. Since $S = S^* \cup S'$, we infer that altogether we have at most $\ell \cdot (|V(G)|^a - 1)(|V(G)|^b - 1)$ choices for S. This shows that $s(G) \leq \ell(|V(G)|^a - 1)(|V(G)|^b - 1)$ and thus $\mathcal{C}_{k,\ell}$ is tame. □

Lemma 4.4.9. *For every two positive integers k and ℓ with $\ell \geq 3$, if the class $\mathcal{C}_{k,\ell-1}$ is tame, then so is the class $\mathcal{C}^*_{k,\ell}$.*

Proof. Fix positive integers k and ℓ with $\ell \geq 3$ and suppose that the class $\mathcal{C}_{k,\ell-1}$ is tame. By Theorem 4.2.3, to show that $\mathcal{C}^*_{k,\ell}$ is tame, it suffices to show that the class $A(\mathcal{C}^*_{k,\ell})$ is tame. We do so by showing that every graph in $A(\mathcal{C}^*_{k,\ell})$ belongs to $\mathcal{C}_{k,\ell-1}$. This will suffice since the class $\mathcal{C}_{k,\ell-1}$ is tame by assumption.

Let $G \in A(\mathcal{C}^*_{k,\ell})$. Then G is an atom that is an induced subgraph of a graph $G^* \in \mathcal{C}^*_{k,\ell}$. By definition of $\mathcal{C}^*_{k,\ell}$, there exist a graph G' in $\mathcal{C}_{k,\ell}$, a clique cover $\{A_1, \ldots, A_\ell\}$ of G', and a minimal separator S^* in the graph $G' - A_\ell$ such that $G^* = G' - S^*$. The fact that S^* is a separator in $G' - A_\ell$ implies that the graph $G' - S^* - A_\ell = G^* - A_\ell$ is disconnected. Thus, there exists a cut-partition (A, B, C) of G^* such that $C = A_\ell$. Note that sets $A \cap V(G)$ and $B \cap V(G)$ are anticomplete to each other in G. Note also that $A_\ell \cap V(G)$ is a (possibly empty) clique in G. Since G has no clique cutset, we infer that one of the sets $A \cap V(G)$ and $B \cap V(G)$ is empty, say $A \cap V(G) = \emptyset$. Furthermore, since the sets A and B are anticomplete to each other in $G^* - A_\ell$, it follows that A is the union of a nonempty subset of the set of cliques $\{A_1 \setminus S^*, \ldots, A_{\ell-1} \setminus S^*\}$. Let $i \in \{1, \ldots, \ell - 1\}$ be such that $A_i \setminus S^* \subseteq A$. The fact that $A \cap V(G) = \emptyset$ now implies that the vertex set of G can be covered with $\ell - 1$ cliques $(A_j \setminus S^*) \cap V(G)$, for $j \in \{1, \ldots, \ell\} \setminus \{i\}$. Since G is an induced subgraph of a graph in $\mathcal{C}_{k,\ell}$, it is L_k-free. Consequently, $G \in \mathcal{C}_{k,\ell-1}$ and the proof is complete. □

Proof of Theorem 4.4.5. Fix a positive integer k. We prove that for every positive integer ℓ, the class $\mathcal{C}_{k,\ell}$ is tame, using induction on ℓ. If $\ell = 1$, then every graph in $\mathcal{C}_{k,\ell}$ is complete and $\mathcal{C}_{k,\ell}$ is tame. If $\ell = 2$, then the class $\mathcal{C}_{k,\ell}$ is tame by Lemma 4.4.7. Suppose now that $\ell \geq 3$ and that the class $\mathcal{C}_{k,\ell-1}$ is tame. By Lemma 4.4.9, the class $\mathcal{C}^*_{k,\ell}$ is tame. Since the classes $\mathcal{C}_{k,\ell-1}$ and $\mathcal{C}^*_{k,\ell}$ are tame, Lemma 4.4.8 implies that so is the class $\mathcal{C}_{k,\ell}$. □

Subclasses of C_4-free graphs

We now derive two consequences of Theorem 4.4.5 dealing with subclasses of C_4-free graphs. Wagon proved in [230] that every $2P_2$-free graph with clique number k is $(k(k+1)/2)$-colorable. This result can be equivalently stated as follows.

Theorem 4.4.10. *For every positive integer k, every $\{kP_1, C_4\}$-free graph has clique cover number at most $k(k-1)/2$.*

Since the graph L_2 is the 4-cycle, this result implies that the class of $\{kP_1, C_4\}$-free graphs is a subclass of the class $\mathcal{C}_{2,k(k-1)/2}$. By Theorem 4.4.5, the class $\mathcal{C}_{2,k(k-1)/2}$ is tame, and we thus obtain the following.

Corollary 4.4.11. *For every positive integer k, the class of $\{kP_1,\ C_4\}$-free graphs is tame.*

The question of which classes of graphs defined by a set of forbidden induced subgraphs with at most four vertices are tame was first investigated in the conference paper [184]. An almost complete dichotomy was obtained, leaving open only two cases: the classes of $\{4P_1, C_4\}$-free and $\{4P_1$, claw, $C_4\}$-free graphs. Clearly, the result of Corollary 4.4.11 resolves both cases. Daniel Lokshtanov kindly communicated to us that the result of Corollary 4.4.11 was also obtained independently (and at approximately the same time) by Peter Gartland. At the same time we managed to generalize Corollary 4.4.11 by replacing kP_1 in the statement of the corollary with $P_2 + kP_1$.

Theorem 4.4.12. *For every positive integer k, the class of $\{P_2+kP_1,\ C_4\}$-free graphs is tame.*

Proof. Let G be a $\{P_2 + kP_1,\ C_4\}$-free graph. Fix a maximum independent set I in G. If $|I| \leq 2k - 1$, then G is $\{2kP_1, C_4\}$-free and Corollary 4.4.11 implies that G has a polynomially bounded number of minimal separators. Thus, in what follows we assume that $|I| \geq 2k$. Let w be a vertex in $V(G) \setminus I$. Since I is a maximal independent set, w has a neighbor in I. Consequently, since G is $(P_2 + kP_1)$-free, w has at most $k - 1$ non-neighbors in I and hence w has at least $|I| - (k - 1) \geq k + 1$ neighbors in I. Suppose next that u and v are two distinct non-adjacent vertices in $V(G) \setminus I$. Since each of u and v has at least $k+1$ neighbors in I, they have at least two common neighbors in I. But this contradicts the fact that G is C_4-free. It follows that $V(G) \setminus I$ is a clique. Since I is an independent set and $V(G) \setminus I$ is a clique, G is $2P_2$-free, and we infer from Lemma 3.1.3 that G has at most $|V(G)|$ minimal separators. □

4.4.2 A dichotomy result

In this section we state and prove the dichotomy result. We will need the following result from the literature, describing the structure of paw-free graphs. A graph G is *complete multipartite* if its vertex set can be partitioned into any number of parts such that two vertices are adjacent if and only if they belong to different parts.

Theorem 4.4.13 (Olariu [188])**.** *A connected paw-free graph G is either C_3-free or complete multipartite.*

If some class of graphs is not tame, then neither is any larger class that contains it. Recall that the class of short prisms, denoted as $\mathcal{L}_2$ is not tame, and that the line graphs of elementary walls are not tame. Then, we infer the following corollary.

Corollary 4.4.14. *The class of $\{3P_1$, diamond$\}$-free graphs and the class of $\{$claw, K_4, C_4, diamond$\}$-free graphs are not tame.*

Proof. Let $G \in \mathcal{L}_2$. Since the vertex set of G is the union of two cliques, G is $3P_1$-free. Moreover, it is not difficult to see that G is diamond-free. Consequently, the class of $\{3P_1$, diamond$\}$-free graphs contains all graphs in $\mathcal{L}_2$ and is therefore not tame by Corollary 4.3.4. Similarly, line graphs of elementary walls are contained in the class of $\{$claw, K_4, C_4, diamond$\}$-free graphs, and by Proposition 4.3.6 the class of $\{$claw, K_4, C_4, diamond$\}$-free graphs is not tame. □

First we prove a number of propositions, each giving a sufficient condition for a family $\mathcal{F}$ of graphs on at most 4 vertices such that the class of $\mathcal{F}$-free graphs is tame. We start with a lemma simplifying the cases with $P_3 + P_1 \in \mathcal{F}$.

Lemma 4.4.15. *Let $\mathcal{F}$ be a family of graphs such that $P_3 + P_1 \in \mathcal{F}$ and let $\mathcal{F}' = (\mathcal{F} \setminus \{P_3 + P_1\}) \cup \{3P_1\}$. Then the class of $\mathcal{F}$-free graphs is tame if and only if the class of $\mathcal{F}'$-free graphs is tame.*

Proof. Let $\mathcal{G}$ and $\mathcal{G}'$ be the classes of $\mathcal{F}$-free and $\mathcal{F}'$-free graphs, respectively. Clearly, every $\mathcal{F}'$-free graph is also $\mathcal{F}$-free (cf. Observation 4.4.1), and hence if $\mathcal{G}$ is tame, then so is $\mathcal{G}'$. Suppose that $\mathcal{G}'$ is tame. By Lemma 4.1.4, there exists an integer $k \geq 0$ such that $s(G) \leq |V(G)|^k - 1$ for all $G \in \mathcal{G}'$. Let $G \in \mathcal{G}$. By Corollary 4.2.2 we may assume that G is coconnected. Since G is coconnected and $(P_3 + P_1)$-free, applying Theorem 4.4.13 to the complement of G implies that G is either a disjoint union of complete graphs, in which case $s(G) \leq 1$, or G is $3P_1$-free, in which case $G \in \mathcal{G}'$ and thus $s(G) \leq |V(G)|^k - 1$. It follows that $\mathcal{G}$ is tame. □

We now consider various families of forbidden induced subgraphs with at most four vertices. We will also need the following result describing the structure of {claw, $C_3 + P_1$}-free graphs. By S_3 we denote the 6-vertex graph obtained from the 6-cycle with vertices $v_1, \ldots, v_6$ in cyclic order by adding to it the chords v_1v_3, v_3v_5, and v_5v_1.

Theorem 4.4.16 (Pouzet et al. [204]). *The class of {claw, C_3+P_1}-free graphs consists of S_3, of the induced subgraphs of $L(K_{3,3})$, of graphs whose connected components are cycles of length at least 4 or paths, and of the complements of these graphs.*

Proposition 4.4.17. *For every $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$, the class of $\{F, C_3 + P_1\}$-free graphs is tame.*

Proof. (1) The class of $\{4P_1, C_3+P_1\}$-free graphs is a subclass of the class of $\{P_2+4P_1, C_3 + P_2\}$-free graphs, which is tame by Theorem 4.4.3.

(2) The class of $\{P_2 + 2P_1, C_3 + P_1\}$-free graphs is a subclass of the class of $\{P_2 + 2P_1, C_3 + P_2\}$-free graphs, which is tame by Theorem 4.4.3.

(3) By Lemma 4.4.15, it suffices to show that the class of $\{3P_1, C_3 + P_1\}$-free graphs is tame. This follows from part (1) of the proposition.

(4) By Corollaries 4.2.1 and 4.2.2, it suffices to prove that the class of connected and coconnected {claw, $C_3 + P_1$}-free graphs is tame. Let G be a connected and coconnected {claw, $C_3 + P_1$}-free graph with at least 10 vertices. By Theorem 4.4.16, G is either a path or a cycle, or the complement of a path or of a cycle. If G is a path or a cycle, then all its minimal separators have size 1 or 2, respectively. If the complement of G is a path or a cycle, then G is $2P_2$-free, and hence has at most $|V(G)|$ minimal separators by Lemma 3.1.3. Thus, in either case the number of minimal separators of G is polynomially bounded. □

Proposition 4.4.18. *The class of $\{P_3 + P_1, C_4\}$-free graphs is tame.*

Proof. Immediate from Lemma 4.4.15 and the fact that the class of $\{3P_1, C_4\}$-free graphs is tame, which follows from Corollary 4.4.11. □

Proposition 4.4.19. *For every $F \in \{4P_1,\ P_2{+}2P_1,\ P_3{+}P_1\}$, the class of $\{F,\ K_4\}$-free graphs is tame.*

Proof. (1) By Ramsey's theorem, the class of $\{4P_1,\ K_4\}$-free graph consists of finitely many graphs, so it is tame.

(2) The class of $\{P_2 + 2P_1,\ K_4\}$-free graphs is a subclass of the class of $\{P_2 + 2P_1,\ K_4 + P_2\}$-free graphs, which is tame by Theorem 4.4.3.

(3) By Lemma 4.4.15, it suffices to show that the class of $\{3P_1,\ K_4\}$-free graphs is tame. This follows from part (1) of the proposition. □

In the proofs of Proposition 4.4.20 and Theorem 4.4.21 we need the fact that the class of P_4-free graphs is tame (stated in Theorem 4.1.5).

Proposition 4.4.20. *For every $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1,\ claw\}$, the class of $\{F,\ paw\}$-free graphs is tame.*

Proof. Let G be an $\{F, \text{paw}\}$-free graph. By Corollary 4.2.1, we may assume that G is connected. Since G is paw-free, Theorem 4.4.13 implies that G is either C_3-free, or complete multipartite. If G is complete multipartite, then G is P_4-free, and thus has a polynomially bounded number of minimal separators by Theorem 4.1.5. Suppose now that G is C_3-free. If $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1\}$, then using the fact that G is K_4-free, Proposition 4.4.19 implies that G has a polynomially bounded number of minimal separators. If F is the claw, then G is a $\{\text{claw}, C_3\}$-free graph, hence G is a path or a cycle, and all its minimal separators have size 1 or 2, respectively. In either case, G has a polynomially bounded number of minimal separators. Thus, the class of $\{F, \text{paw}\}$-free graphs is tame. □

Finally, we obtain the characterization of tame graph classes defined by the set of forbidden induced subgraphs on at most four vertices. To appreciate our result (Theorem 4.4.21), note that up to isomorphism, there are 11 four-vertex graphs, which means that there are $2^{11} = 2048$ different graph classes defined by a set of forbidden induced subgraphs with exactly four vertices, and even more graph classes defined by a set of forbidden induced subgraphs with at most four vertices.

In Fig. 4.6 we give an overview of maximal tame and minimal non-tame classes of $\mathcal{F}$-free graphs, where $\mathcal{F}$ contains graphs with at most four vertices. A similar figure with respect to the boundedness of the clique-width can be found in [45].

Theorem 4.4.21. *For every family $\mathcal{F}$ of graphs with at most 4 vertices, the following statements are equivalent.*

1. *The class of $\mathcal{F}$-free graphs is tame.*

2. *The class of $\mathcal{F}$-free graphs does not contain any of the following graph classes: the class of $\{C_3,\ C_4\}$-free graphs, the class of $\{3P_1,\ diamond\}$-free graphs, and the class of $\{claw,\ K_4,\ C_4,\ diamond\}$-free graphs.*

3. *$\mathcal{F} \trianglelefteq \mathcal{F}'$ for at least one of the following families $\mathcal{F}'$:*

   (a) $\mathcal{F}' = \{P_4\}$,

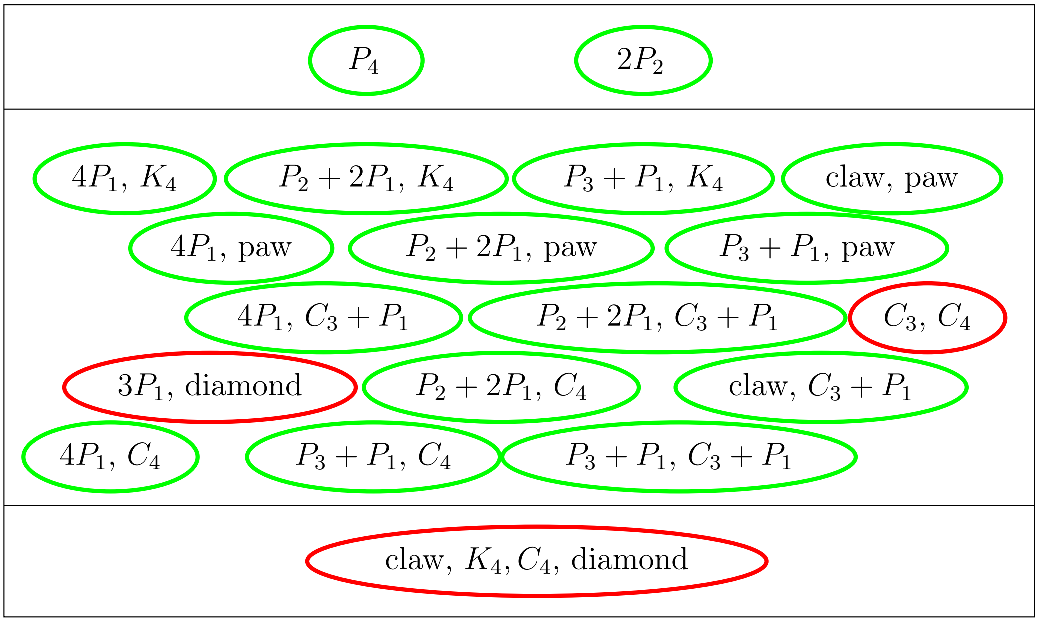


Figure 4.6: Overview of the dichotomy result. Maximal tame classes correspond to sets $\mathcal{F}$ of forbidden induced subgraphs depicted in green ellipses, while minimal non-tame classes correspond to sets depicted in red ellipses (in brighter, resp., darker ellipses in gray-scale printing).

(b) $\mathcal{F}' = \{2P_2\}$,

(c) $\mathcal{F}' = \{F,\ paw\}$ *for some* $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1,\ claw\}$,

(d) $\mathcal{F}' = \{F,\ C_3 + P_1\}$ *for some* $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1,\ claw\}$,

(e) $\mathcal{F}' = \{F,\ K_4\}$ *for some* $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1\}$,

(f) $\mathcal{F}' = \{F,\ C_4\}$ *for some* $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1\}$.

Proof. Let $\mathcal{F}$ be a family of graphs on at most 4 vertices.

Suppose first that the class of $\mathcal{F}$-free graphs is tame and, for a contradiction, that the class of $\mathcal{F}$-free graphs contains the class of $\mathcal{F}'$-free graphs for some $\mathcal{F}' \in \{\{C_3,\ C_4\}, \{3P_1,\ \text{diamond}\}, \{\text{claw},\ K_4,\ C_4,\ \text{diamond}\}\}$. If $\mathcal{F}' = \{C_3,\ C_4\}$, then the class of $\mathcal{F}'$-free graphs is not tame by Proposition 4.4.2. If $\mathcal{F}' = \{3P_1,\ \text{diamond}\}$, then the class of $\mathcal{F}'$-free graphs is not tame by Corollary 4.4.14. Note that the line graphs of elementary walls are $\{\text{claw},\ K_4,\ C_4,\ \text{diamond}\}$-free. If $\mathcal{F}' = \{\text{claw},\ K_4,\ C_4,\ \text{diamond}\}$, then Proposition 4.3.6 implies that the class of $\mathcal{F}'$-free graphs is tame. It follows by Observation 4.1.2 that the class of $\mathcal{F}$-free graphs is not tame, a contradiction. Thus, the first statement implies the second one.

Suppose now that for all $\mathcal{F}' \in \{\{C_3,\ C_4\}, \{3P_1,\ \text{diamond}\}, \{\text{claw},\ K_4,\ C_4,\ \text{diamond}\}\}$, the class of $\mathcal{F}$-free graphs does not contain the class of $\mathcal{F}'$-free graphs. We want to prove that $\mathcal{F} \trianglelefteq \mathcal{F}'$ where $\mathcal{F}'$ satisfies one of the following: $\mathcal{F}' = \{P_4\}$, $\mathcal{F}' = \{2P_2\}$, $\mathcal{F}' = \{F,\ \text{paw}\}$ for some $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1,\ \text{claw}\}$, $\mathcal{F}' = \{F,\ C_3 + P_1\}$ for some $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1,\ \text{claw}\}$, $\mathcal{F}' = \{F,\ K_4\}$ for some $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1\}$, or $\mathcal{F}' = \{F,\ C_4\}$ for some $F \in \{4P_1,\ P_2 + 2P_1,\ P_3 + P_1\}$.

If some graph $F \in \mathcal{F}$ is an induced subgraph of $2P_2$ or of P_4, then $\mathcal{F} \trianglelefteq \mathcal{F}'$ for $\mathcal{F}' = \{2P_2\}$ or $\mathcal{F}' = \{P_4\}$. Thus, from now on we assume that no graph $F \in \mathcal{F}$ is an induced subgraph of either $2P_2$ or P_4. Let $A = \{C_3,\ C_4,\ C_3 + P_1,\ \text{paw},\ \text{diamond},\ K_4\}$

and $B = \{3P_1, 4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$. Note that every member of B contains $3P_1$ as an induced subgraph, and similarly every member of A contains C_3 or C_4 as an induced subgraph. Since every graph in $\mathcal{F}$ has at most 4 vertices and is not an induced subgraph of either $2P_2$ or P_4, we infer that $\mathcal{F} \subseteq A \cup B$. Let $A' = A \setminus \{\text{diamond}\}$. If $\mathcal{F} \cap A' = \emptyset$, then $\mathcal{F} \subseteq \{\text{diamond}\} \cup B$ and the class of $\mathcal{F}$-free graphs contains the class of $\{3P_1, \text{diamond}\}$-free graphs, a contradiction. It follows that $\mathcal{F} \cap A' \neq \emptyset$. If $\mathcal{F} \cap B = \emptyset$, then $\mathcal{F} \subseteq A$ and the class of $\mathcal{F}$-free graphs contains the class of $\{C_3, C_4\}$-free graphs, a contradiction. Therefore, $\mathcal{F} \cap B \neq \emptyset$. If $\mathcal{F} \cap \{C_3, C_3 + P_1, \text{paw}\} \neq \emptyset$, then the fact that $\mathcal{F} \cap B \neq \emptyset$ implies that $\mathcal{F} \trianglelefteq \mathcal{F}'$ where $\mathcal{F}' = \{F, \text{paw}\}$ or $\mathcal{F}' = \{F, C_3 + P_1\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$. Assume now that $\mathcal{F} \cap \{C_3, C_3 + P_1, \text{paw}\} = \emptyset$. As $\mathcal{F} \cap A' \neq \emptyset$, it follows that $\mathcal{F} \cap \{C_4, K_4\} \neq \emptyset$. Let $B' = B \setminus \{\text{claw}\}$. If $\mathcal{F} \cap B' = \emptyset$, then $\mathcal{F} \subseteq \{\text{claw}, C_4, K_4, \text{diamond}\}$ and the class of $\mathcal{F}$-free graphs contains the class of $\{\text{claw}, K_4, C_4, \text{diamond}\}$-free graphs, a contradiction. It follows that $\mathcal{F} \cap B' \neq \emptyset$. Therefore, using that $\mathcal{F} \cap \{C_4, K_4\} \neq \emptyset$, we infer that $\mathcal{F} \trianglelefteq \mathcal{F}'$ where $\mathcal{F}' = \{F, C_4\}$ or $\mathcal{F}' = \{F, K_4\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1\}$. Thus, the second statement implies the third one.

Finally, suppose that $\mathcal{F} \trianglelefteq \mathcal{F}'$ where $\mathcal{F}'$ satisfies one of the following: $\mathcal{F}' = \{P_4\}$, $\mathcal{F}' = \{2P_2\}$, $\mathcal{F}' = \{F, \text{paw}\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$, $\mathcal{F}' = \{F, C_3 + P_1\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$, $\mathcal{F}' = \{F, K_4\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1\}$, or $\mathcal{F}' = \{F, C_4\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1\}$. Note that the class of $\mathcal{F}$-free graphs is contained in the class of $\mathcal{F}'$-free graphs, hence by Observation 4.1.2 it suffices to show that the class of $\mathcal{F}'$-free graphs is tame. If $\mathcal{F}' = \{P_4\}$ or $\mathcal{F}' = \{2P_2\}$, then the class of $\mathcal{F}'$-free graphs is tame by Theorem 4.1.5 and Corollary 4.1.6, respectively. If $\mathcal{F}' = \{F, \text{paw}\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$, then the class of $\mathcal{F}'$-free graphs is tame by Proposition 4.4.20. If $\mathcal{F}' = \{F, C_3 + P_1\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1, \text{claw}\}$, then the class of $\mathcal{F}'$-free graphs is tame by Proposition 4.4.17. If $\mathcal{F}' = \{F, K_4\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1\}$, then the class of $\mathcal{F}'$-free graphs is tame by Proposition 4.4.19. Finally, if $\mathcal{F}' = \{F, C_4\}$ for some $F \in \{4P_1, P_2 + 2P_1, P_3 + P_1\}$, then the class of $\mathcal{F}'$-free graphs is tame by Corollary 4.4.11 (if $F = 4P_1$), by Proposition 4.4.18 (if $F = P_3 + P_1$), or by Theorem 4.4.12 (if $F = P_2 + 2P_1$). This shows that the third statement implies the first one and completes the proof. □

4.5 Characterization of tame graph classes with a forbidden induced minor or induced topological minor

The results presented so far in this chapter are restricted to graph classes that are closed under induced subgraphs. However, outside the realm of tame graph classes, many other graph inclusion relations—for example, the minor, topological minor, subgraph, induced minor, and induced topological minor relations—have been studied in the literature and have proved important in various contexts. In particular, dichotomy results for various properties were developed for graph classes defined by a single excluded graph with respect to one of the above relations. Such properties include bounded clique-width [20], well-quasi ordering [28, 94], equivalence of bounded treewidth and bounded clique number [86], bounded tree-independence number [88], and polynomial-time solvability of Graph Isomorphism [189].

Motivated by this state of the art, we focus in this section on the two remaining "induced" relations, the induced minor and induced topological minor relation. These relations are more challenging to work with than the induced subgraph relation. For example, there exist graphs H such that detecting H as an induced minor or an induced topological minor is NP-complete. For the induced minor relation, this was shown by Fellows, Kratochvíl, Middendorf, and Pfeiffer in 1995 [104]; recently, Korhonen and Lokshtanov showed that this can happen even if H is a tree [158]. For the induced topological minor relation, Lévêque, Lin, Maffray and Trotignon [163] and Maffray, Trotignon and Vušković [170] showed that the problem is NP-complete if H is the complete graph K_5, or the complete bipartite graph $K_{2,4}$, respectively. In general, the following two questions are widely open.

Question 4.5.1. For which graphs H there exists a polynomial-time algorithm for determining if a given graph G contains H as an induced minor?

Question 4.5.2. For which graphs H there exists a polynomial-time algorithm for determining if a given graph G contains H as an induced topological minor?

In both cases, the problem is solvable in polynomial time if every component of H is a path, since in this case it suffices to check if H is present as an induced subgraph. Furthermore, if H is a graph with at most four vertices, then determining if a given graph G contains H as an induced minor can be done in polynomial time (see [85,134]). This is also the case if H is the complete bipartite graph $K_{2,3}$ (see [85]). Also for the induced topological minor relation, only few polynomial cases are known (see [65, 161, 161]).

Graph classes excluding a fixed planar graph H as an induced minor are also relevant for the complexity of Maximum Weight Independent Set (MWIS): Given a graph G and a vertex weight function $w : V(G) \to \mathbb{Q}_+$, compute an independent set I in G maximizing its weight $\sum_{x\in I} w(x)$. The problem of determining the computational complexity of MWIS in particular graph classes has been extensively studied. In particular, the problem is known to be NP-hard in the class of planar graphs (see [117]), which implies that MWIS remains NP-hard in graph classes defined by a single forbidden induced minor H when H is nonplanar. For the case when a planar graph H is forbidden as an induced minor, Dallard, Milanič, and Štorgel posed the following question in [88].

Question 4.5.3. Is MWIS solvable in polynomial time in the class of H-induced-minor-free graphs for every planar graph H?

This question is still open, even for the cases when H is the path P_7 or the cycle C_6, but some partial results are known. Question 4.5.3 has an affirmative answer for all of the following graphs H: the path P_6 (as shown by Grzesik et al. [130]), the cycle C_5 (as shown by Abrishami et al. [4]), the complete bipartite graph $K_{2,t}$ for any positive integer t, as well as graphs obtained from the complete graph K_5 by deleting either one edge or two disjoint edges (as shown by Dallard, Milanič, and Štorgel [88]), and the t-friendship graph (that is, t disjoint edges plus a vertex fully adjacent to them; as shown by Bonnet et al. [37][1]). Quasi-polynomial-time algorithms are also known for the cases when H is either a path (as shown by Gartland and Lokshtanov [119] and

[1]The paper [37] solves the unweighted version of the problem, however, the methods can be easily extended to the weighted case.

Pilipczuk et al. [199]) or, more generally, a cycle (as shown by Gartland et al. [122]), or the graph $tC_3 + C_4$ for any integer $t \geq 0$ (as shown by Bonnet et al. [37]). Furthermore, Korhonen showed that for any planar graph H, MWIS can be solved in subexponential time in the class of H-induced-minor-free graphs [157].

Recall that if $\mathcal{G}$ is a tame graph class, then MWIS and many other problems are solvable in polynomial time for graphs in $\mathcal{G}$. This motivates the following questions.

Question 4.5.4. For which graphs H is the class of graphs excluding H as an induced minor tame?

Question 4.5.5. For which graphs H is the class of graphs excluding H as an induced topological minor tame?

In particular, this may provide further partial answers to Question 4.5.3. Let us remark that for the case of induced subgraph relation, the aforementioned dichotomy of hereditary graph classes defined by a finite set of forbidden induced subgraphs into tame and feral (see [115, 120]) implies that a graph class defined by a *single* forbidden induced subgraph H is tame if H is an induced subgraph of the path P_4 or of the graph $2P_2$, and feral, otherwise.

In this section we completely answer Questions 4.5.4 and 4.5.5. We show that every graph class defined by a single forbidden induced minor or induced topological minor is either tame or feral, and classify the two cases. Graphs used in our characterizations are depicted in Fig. 4.7.

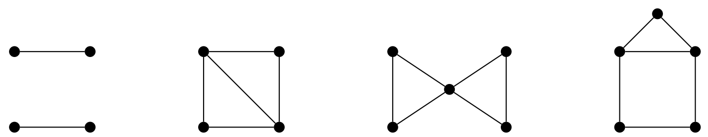


Figure 4.7: From left to right: the $2P_2$, the diamond, the butterfly, and the house.

Theorem 4.5.6. *Let H be a graph and let $\mathcal{G}$ be the class of graphs that do not contain H as an induced minor. Then, the following statements are equivalent:*

(1) $\mathcal{G}$ is tame.

(2) $\mathcal{G}$ is not feral.

(3) H is an induced subgraph of the diamond, the butterfly, or the house.

(4) H is an induced minor of the butterfly or of the house.

Theorem 4.5.7. *Let H be a graph and let $\mathcal{G}$ be the class of graphs that do not contain an induced subdivision of H. Then, the following statements are equivalent:*

(1) $\mathcal{G}$ is tame.

(2) $\mathcal{G}$ is not feral.

(3) H is an induced subgraph of $2P_2$, the diamond, or the house.

(4) H is an induced topological minor of $2P_2$ or of the house.

Then, we complement the above results by analyzing the complexity of the recognition problems for the maximal tame graph classes in each of the above two theorems. These correspond to Question 4.5.4 for the case when H is either the butterfly or the house and to Question 4.5.5 for the case when H is either $2P_2$ or the house. As already observed, determining if a given graph G contains $2P_2$ as an induced topological minor can be done in polynomial time. Determining if a given graph G contains the butterfly as an induced minor can also be done in polynomial time, using a characterization of such graphs due to Dumas and Hilaire (personal communication, 2024; for completeness, we present the argument in Section 4.6). We provide polynomial-time algorithms for the remaining two cases.

Theorem 4.5.8. *Determining if a given graph G contains the house as an induced minor can be done in time $\mathcal{O}(n^8 m \log^2 n)$.*

Theorem 4.5.9. *Determining if a given graph G contains the house as an induced topological minor can be done in time $\mathcal{O}(n^8 m \log^2 n)$.*

Applying the result by Fomin, Todinca, and Villanger from Theorem 3.1.1 to the two maximal tame graph classes identified in Theorems 4.5.6 and 4.5.7 yields the following algorithmic implications of our results.

Corollary 4.5.10. *For fixed integer $t \geq 0$, fixed CMSO_2 formula ϕ, the (t, ϕ)-*MAXIMUM WEIGHT INDUCED SUBGRAPH *problem is solvable by a robust polynomial-time algorithm whenever the input graph does not contain the butterfly as an induced minor or the house as an induced topological minor.*

The above graph classes generalize some graph classes extensively studied in the literature (see, e.g., [47, 126, 219]). First, the class of graphs that do not contain the butterfly as an induced minor generalizes the class of $2P_2$-free graphs (hence, in particular, the classes of split graphs and complements of chordal graphs). Second, the class of graphs that do not contain the butterfly as an induced minor (or, more generally, as an induced topological minor) is a common generalization of the classes of chordal graphs and cographs.

Since MAXIMUM WEIGHT INDEPENDENT SET is a special case of (t, ϕ)-MAXIMUM WEIGHT INDUCED SUBGRAPH, Corollary 4.5.10 provides further partial support for Question 4.5.3, giving an affirmative answer to the question for the cases when H is either the butterfly or the house. Let us also note that the butterfly is the 2-friendship graph, hence, the former result also follows from the aforementioned result of Bonnet et al. [37].

4.5.1 Preliminary results

Given a graph G, a *walk* in G is a finite nonempty sequence $W = (w_1, \ldots, w_k)$ of vertices in G such that every two consecutive vertices are joined by an edge in G. If $w_1 = w_k$, then W is said to be a *closed* walk. The *length* of a walk $(w_1, \ldots, w_k)$ is defined to be $k - 1$. Given two vertices u, v in a graph G, a *u,v-walk* in G is any walk $(w_1, \ldots, w_k)$ in G such that $u = w_1$ and $v = w_k$. More generally, for two sets $A, B \subseteq V(G)$, *a walk from A to B* is any walk $(w_1, \ldots, w_k)$ in G such that $w_1 \in A$ and $w_k \in B$. Given two walks $W = (w_1, \ldots, w_k)$ and $Z = (z_1, \ldots, z_\ell)$ in a graph G such that $w_k = z_1$, we define the *concatenation* of W and Z to be the walk obtained by traversing first W and then Z, that is, the walk $(w_1, \ldots, w_k = z_1, z_2, \ldots, z_\ell)$. The concatenation of walks W and Z will be denoted by $W \oplus Z$. A *path* in G is a walk in which all vertices are pairwise distinct. Given a u, v-path P, the vertices u and v are the *endpoints* of P, while all the other vertices are its *internal vertices*. A *cycle* in G is a closed walk with length at least 3 such that all its vertices are pairwise distinct, except that $w_1 = w_k$.

Recall that any induced topological minor of a graph is also an induced minor of the same graph, so given a graph H it holds that a class of H-induced-minor-free graphs is a subclass of the class of graphs that do not contain H as an induced topological minor. Recall also that a graph H is an induced minor of a graph G if and only if there is an induced minor model $\{X_v\}_{v \in V(H)}$ of H in G. In the rest of this section we establish a general result about induced minor models of graphs containing vertices of degree 2.

We identify a property of induced minor models that can be assumed without loss of generality whenever the pattern graph contains an edge-disjoint collection of particular walks. A walk $W = (w_1, \ldots, w_k)$ in a graph H is said to be a *thin walk* if all the vertices of W are pairwise distinct, except that possibly $w_1 = w_k$, and for all $i \in \{2, \ldots, k-1\}$, vertex w_i has degree 2 in H. Given a thin walk $W = (w_1, \ldots, w_k)$ in a graph H, vertices $w_2, \ldots, w_{k-1}$ will be referred to as the *internal vertices* of W.

Lemma 4.5.11. *Let H be a graph, let W be a thin walk in H, and let H be an induced minor of a graph G. Then, there exists an induced minor model $\{X_v\}_{v \in V(H)}$ of H in G such that $|X_w| = 1$ for every internal vertex w of W.*

Proof. Let $\{X_v\}_{v \in V(H)}$ be an induced minor model of H in G. Let $W = (w_1, \ldots, w_k)$. For simplicity, write X_i for X_{w_i} for all $i \in \{1, \ldots, k\}$. Note that the sets $X_1, \ldots, X_k$ are nonempty and pairwise disjoint, except that $X_1 = X_k$ if W is a closed walk. If $k = 2$, there is nothing to show, so we assume that $k \geq 3$. Consider the following particular way of constructing a walk in G from X_1 to X_k. For all $i \in \{1, \ldots, k-1\}$, let $u_i v_{i+1}$ be an edge in G such that $u_i \in X_i$ and $v_{i+1} \in X_{i+1}$. Moreover, for all $i \in \{2, \ldots, k-1\}$, let P^i be a v_i,u_i-path in $G[X_i]$. Then, the following concatenation of walks

$$(u_1, v_2) \oplus \bigoplus_{i=2}^{k-1} \big(P^i \oplus (u_i, v_{i+1})\big)$$

is a walk in G from X_1 to X_k with all vertices pairwise distinct, except that the endpoints might coincide. Any walk that can be obtained by the above procedure will be referred to as a *W-monotone* walk in G.

Let $Z = (z_1, \ldots, z_r)$ be a shortest W-monotone walk in G. Note that the vertices $z_2, \ldots, z_{r-1}$ all belong to $\bigcup_{i=2}^{k-1} X_i$. The minimality of Z and the fact that all internal vertices of W have degree 2 in H imply that the vertices $z_2, \ldots, z_{r-1}$ have degree 2 in the subgraph of G induced by $\{z_1, \ldots, z_r\}$. We now modify $\{X_v\}_{v \in V(H)}$ to obtain an alternative induced minor model $\{Y_v\}_{v \in V(H)}$ of H in G, as follows:

- $Y_v = X_v$ for all $v \in V(H) \setminus \{w_1, \ldots, w_k\}$,
- $Y_{w_i} = \{z_i\}$ for all $i \in \{2, \ldots, k-1\}$,
- $Y_{w_k} = X_k \cup \{z_i \colon k \leq i < r\}$, and

- if $w_1 \neq w_k$, then $Y_{w_1} = X_1$.[2]

Note that $|Y_{w_i}| = 1$ for all $i \in \{2, \ldots, k-1\}$ and for each $v \in V(H)$, the subgraph of G induced by Y_v is connected. Furthermore, the aforementioned properties of the subgraph of G induced by $\{z_1, \ldots, z_r\}$ imply that $\{Y_v\}_{v \in V(H)}$ is indeed an induced minor model of H in G. □

A repeated application of Lemma 4.5.11 and its proof leads to the following.

Proposition 4.5.12. *Let H be a graph, $\mathcal{W}$ be a set of edge-disjoint thin walks in H, and U be the set of all internal vertices of walks in $\mathcal{W}$. Let H be an induced minor of a graph G. Then, there exists an induced minor model $\{X_v\}_{v \in V(H)}$ of H in G such that $|X_u| = 1$ for all $u \in U$.*

Proof. Let $\{X_v\}_{v \in V(H)}$ be an induced minor model of H in G. Consider a walk $W = (w_1, \ldots, w_k)$ in $\mathcal{W}$. Modifying the induced minor model $\{X_v\}_{v \in V(H)}$ as described in the proof of Lemma 4.5.11 yields an induced minor model $\{Y_v\}_{v \in V(H)}$ of H in G such that $|Y_u| = 1$ for every internal vertex u in W and, moreover, $\bigcup_{i=2}^{k-1} Y_{p_i} \subseteq \bigcup_{i=2}^{k-1} X_{p_i}$. In particular, for any walk $W' \in \mathcal{W}$ such that $W' \neq W$ this procedure does not interfere with sets X_u for internal vertices u of W. Thus, since no internal vertex of any walk $\mathcal{W}$ belongs to any other walk in $\mathcal{W}$, we can apply the construction to all walks in $\mathcal{W}$ in any order, resulting in an induced minor model $\{Z_v\}_{v \in V(H)}$ of H in G such that $|Z_u| = 1$ for all $u \in U$. □

4.5.2 Sufficient conditions for tameness

In this section we identify new tame classes of graphs defined by a single excluded induced minor or induced topological minor. Some result of this section will make use of the following result by Gajarský et al. [115]. Note that a k-skinny ladder is a graph consisting of two induced anti-adjacent paths $P = (p_1, \ldots, p_k)$, $Q = (q_1, \ldots, q_k)$, an independent set $R = (r_1, \ldots, r_k)$, and edges $\bigcup_{i=1}^{k} \{p_i r_i, q_i r_i\}$

Theorem 4.5.13 (Theorem 3 in [115]). *For every positive integer k there exists a polynomial p of degree $\mathcal{O}(k^3 \cdot (8k^2)^{k+2})$ such that every graph G that is k-creature-free and does not contain a k-skinny ladder as an induced minor contains at most $p(|V(G)|)$ minimal separators.*

By Theorem 4.5.13, in order to prove that a graph class $\mathcal{G}$ is tame, it suffices to prove that there exists an integer k such that every graph in $\mathcal{G}$ is k-creature-free and does not contain k-skinny ladder as an induced minor. We show that if $\mathcal{G}$ is the class of graphs excluding the butterfly as an induced minor or the class of graphs excluding the house as an induced topological minor (see Fig. 4.8), then the above condition is satisfied with $k = 3$. These results will be crucial for the dichotomy theorems developed in Section 4.5.3.

Figure 4.8: The butterfly (left) and the house (right).

[2]Note that if $w_1 = w_k$, then $Y_{w_1} = Y_{w_k}$.

We start with the easy case of graphs excluding a butterfly as an induced minor.

Lemma 4.5.14. *Every butterfly-induced-minor-free graph is 3-creature-free.*

Proof. Let G be a butterfly-induced-minor-free graph. Suppose for a contradiction that there exists a 3-creature $H = (A, B, X, Y)$ in G, with $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2, y_3\}$. Note that each vertex in $X = \{x_1, x_2, x_3\}$ has a neighbor in A, and each vertex in $Y = \{y_1, y_2, y_3\}$ has a neighbor in B. Contracting $A \cup B \cup \{x_2, y_2\}$ in H to a single vertex x gives a butterfly graph induced by vertices x, x_1, y_1, x_3, y_3, a contradiction. □

Lemma 4.5.15. *Let G be a butterfly-induced-minor-free graph. Then, G does not contain the 3-skinny ladder as an induced minor.*

Proof. Suppose for a contradiction that the 3-skinny ladder graph H is an induced minor of G and let $\{X_v\}_{v \in V(H)}$ be an induced minor model of H in G. Let $p_1, p_2, p_3, q_1, q_2, q_3, r_1, r_2, r_3$ be the vertices of H, as in the definition of k-skinny ladder. By Proposition 4.5.12 we may assume that $X_{p_i} = \{p_i\}$ and $X_{q_i} = \{q_i\}$ for $i \in \{1, 3\}$ and $X_{r_j} = \{r_j\}$ for all $j \in \{1, 2, 3\}$. Contracting the set $X_{q_2} \cup X_{p_2} \cup \{p_1, p_3, r_2\}$ into a single vertex x results in a butterfly graph induced by $\{x, r_1, q_1, r_3, q_3\}$, a contradiction. □

Lemmas 4.5.14 and 4.5.15 and Theorem 4.5.13 immediately imply the following.

Theorem 4.5.16. *The class of butterfly-induced-minor-free graphs is tame.*

To derive a similar conclusion for the class of graphs excluding a house as an induced topological minor, we build on a recent work of Dallard et al. [85] who studied the class of $K_{2,3}$-induced-minor-free graphs and obtained a polynomial-time recognition algorithm for graphs in the class. As part of their approach, they described a family $\mathcal{F}$ of graphs such that a graph G contains $K_{2,3}$ as an induced minor if and only if G contains a member of $\mathcal{F}$ as an induced subgraph. We omit the exact description of the family $\mathcal{F}$, as we will not need it; however, a lemma from Dallard et al. [85] used in their proof will be useful for us.

To explain the lemma, we need to introduce three more specific graph classes (see Fig. 4.9). Let $\mathcal{S}$ be the class of subdivisions of the claw. The class $\mathcal{T}$ is the class of graphs that can be obtained from three paths of length at least one by selecting one endpoint of each path and adding three edges between those endpoints so as to create a triangle. The class $\mathcal{M}$ is the class of graphs H that consist of a path P and a vertex a, called the *center* of H, such that a is non-adjacent to the endpoints of P and a has at least two neighbors in P. Given a graph $H \in \mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$, the *extremities* of H are the vertices of degree one as well as the center of H in case $H \in \mathcal{M}$. Observe that any graph $H \in \mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$ has exactly three extremities.

Lemma 4.5.17 (Dallard et al. [85])**.** *Let G be a graph and I be an independent set in G with $|I| = 3$. If there exists a connected component C of $G - I$ such that $I \subseteq N(C)$, then there exists an induced subgraph H of $G[N[C]]$ such that $H \in \mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$ and I is exactly the set of extremities of H.*

Using Lemma 4.5.17 we now derive the following.

Lemma 4.5.18. *Let G be a graph that does not contain an induced subdivision of the house. Then, G is 3-creature-free.*

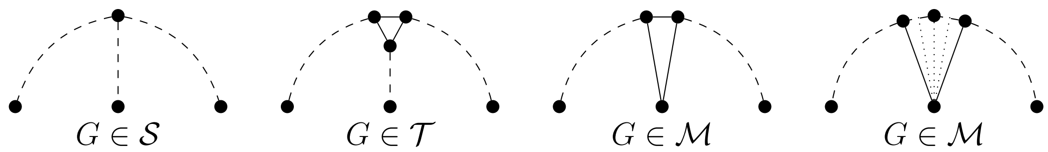


Figure 4.9: A schematic representation of graphs in $\mathcal{S}$, $\mathcal{T}$, and $\mathcal{M}$. Dashed edges represent paths of positive length. Dotted edges may be present or not. The figure is adapted from [85].

Proof. Suppose for a contradiction that there exists a 3-creature $H = (A, B, X, Y)$ in G, with $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2, y_3\}$. Note that each vertex in $X = \{x_1, x_2, x_3\}$ has a neighbor in A. Similarly, each vertex in $Y = \{y_1, y_2, y_3\}$ has a neighbor in B. Let Q_A (resp., Q_B) be a shortest x_1, x_2-path (resp., y_1, y_2-path) of length at least two in G such that all its internal vertices belong to A (resp., B).

Assume that there exists an edge in X. By symmetry, we may assume that x_1x_2 is an edge. If $y_1y_2 \in E(G)$, then the vertices $\{y_1, y_2\}$ together with the vertices from the path Q_A induce a subdivision of the house in G. If $y_1y_2 \notin E(G)$, then the vertices from the paths Q_A and Q_B induce a subdivision of the house in G. Both cases lead to a contradiction, hence, X is an independent set in G, and, by symmetry, so is Y.

Note that A and B induced connected components of $H-X$ and $H-Y$, respectively, and $X \subseteq N_H(A)$ and $Y \subseteq N_H(B)$. Hence, by Lemma 4.5.17, there exist induced subgraphs H_A and H_B of $H[N[A]]$ and $H[N[B]]$, respectively, such that $H_A, H_B \in \mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$ and and X and Y are exactly the sets of extremities of H_A and H_B, respectively.

Assume first that one of these two graphs belongs to $\mathcal{T}$. Up to symmetry, we may assume that $H_A \in \mathcal{T}$. Then, the vertices x_1, x_2, x_3 connect to one triangle in H_A. It follows that the shortest x_1, x_2-path in H_A together with the remaining vertex of the triangle, the edges x_1y_1, x_2y_2, and the path Q_B form an induced subdivision of the house in G, a contradiction. We may therefore assume that neither H_A nor H_B belong to $\mathcal{T}$. Hence, $H_A, H_B \in \mathcal{S} \cup \mathcal{M}$.

Assume next that both graphs H_A and H_B belong to $\mathcal{S}$. In this case H and, hence, G contains an induced subgraph of the form $\Gamma_{i,j,k}$ with $3 \leq i \leq j \leq k$, which is a subdivision of the house, yielding a contradiction. Therefore, by symmetry, we may assume that the graph H_A does not belong to $\mathcal{S}$. It follows that $H_A \in \mathcal{M}$.

Recall that an arbitrary graph in $\mathcal{M}$ consists of a path P and a center a such that a is non-adjacent to the endpoints of P and a has at least two neighbors in P. Up to symmetry, we may assume that x_1 is the center in H_A. Let a be the neighbor of x_1 in H_A closest to x_2 and let a' be a neighbor of x_1 in the same graph such that x_1 has no neighbors in the interior of the path in A from a to a'. Let P be the x_2, a'-path in the graph $H_A - x_1$. We now analyze two cases depending on the structure of H_B; recall that $H_B \in \mathcal{S} \cup \mathcal{M}$.

Assume first that H_B belongs to $\mathcal{S}$. Then, the path P and the x_1, x_2-path in H_B, together with the edges x_1a, x_1a' form an induced subdivision of the house in G, a contradiction.

Consider now the case when H_B belongs to $\mathcal{M}$. Up to symmetry, there are two subcases: either x_1 is the center of H_B, or x_2 is the center of H_B. Assume first that x_1 is the center of H_B. Let b be the neighbor of x_1 in H_B that is closest to x_2. Then, the path P and the x_2, b-path in H_B together with the edges x_1a, x_1a', x_1b form an induced subdivision of the house in G, a contradiction. Finally, assume that x_2 is the center of H_B. Let b be the neighbor of x_2 in the path $H_B - x_2$ that is closest to x_1. Then, the path P and the x_1, b-path in the graph H_B together with the edges x_1a, x_1a', x_2b form an induced subdivision of the house in G, a contradiction. □

It remains to prove that every graph that does not contain any induced subdivision of the house excludes the 3-skinny ladder as an induced minor. Recall the notation of graphs $\Gamma_{i,j,k}$ from Chapter 2 and note that the 3-skinny ladder is isomorphic to the graph $\Gamma_{2,4,4}$ (see Fig. 4.10).

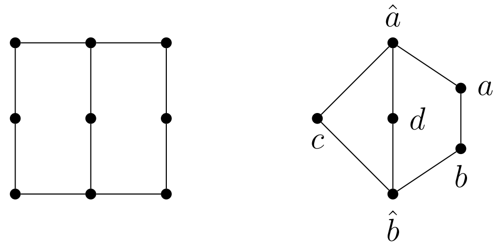


Figure 4.10: The graphs $\Gamma_{2,4,4}$ (left) and $\Gamma_{2,2,3}$ (right).

Note also that the graph $\Gamma_{2,2,3}$ is an induced minor of the graph $\Gamma_{2,4,4}$. Hence, in order to show that every graph that does not contain any induced subdivision of the house excludes the 3-skinny ladder as an induced minor, it suffices to show the following.

Lemma 4.5.19. *Let G be a graph that does not contain an induced subdivision of the house. Then, G is $\Gamma_{2,2,3}$-induced-minor-free.*

Proof. Suppose for a contradiction that G contains $\Gamma_{2,2,3}$ as an induced minor. Note that the three paths in $\Gamma_{2,2,3}$ joining the two vertices of degree three form a collection of edge-disjoint thin walks in $\Gamma_{2,2,3}$. Hence, using the notation from Fig. 4.10, we may assume by Proposition 4.5.12 that there exists an induced minor model $\{X_v\}_{v \in V(\Gamma_{2,2,3})}$ of $\Gamma_{2,2,3}$ in G such that $X_u = \{u\}$ for all $u \in \{a, b, c, d\}$. Denoting by H the subgraph of G induced by $\{X_v\}_{v \in V(\Gamma_{2,2,3})}$, we have that $V(H) = \{a, b, c, d\} \cup A \cup B$, where both A and B induce connected subgraphs of H, vertices c and d have neighbors in both A and B, vertex a has neighbors in A but not in B, vertex b has neighbors in B but not in A, and ab is an edge in H, and the sets $\{a, c, d\}$ and $\{b, c, d\}$ are independent in H.

Note that A is a connected component of $H - S_A$ such that each vertex in $S_A = \{a, c, d\}$ has a neighbor in A. Similarly, B in a connected component of $H - S_B$ such that each vertex in $S_B = \{b, c, d\}$ has a neighbor in B. By Lemma 4.5.17, there exist induced subgraph H_A of $H[N[A]]$ and H_B of $H[N[B]]$, such that $H_A, H_B \in \mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$ and S_A and S_B are exactly the sets of extremities of H_A and H_B, respectively.

Assume first that one of these two graphs belongs to $\mathcal{T}$. Up to symmetry, we may assume that $H_A \in \mathcal{T}$. Then, the vertices a, c, d connect to one triangle in A. Let P be a shortest c, d-path with interior in B. It follows that the shortest c, d-path in H_A, together with the remaining vertex of the triangle and the path P induces a subdivision of the house in G. We may therefore assume that neither H_A nor H_B belong to $\mathcal{T}$. Hence, $H_A, H_B \in \mathcal{S} \cup \mathcal{M}$.

Assume next that both of these graphs belong to $\mathcal{S}$. In this case G contains an induced subgraph of the form $\Gamma_{i,j,k}$ with $2 \leq i \leq j \leq k$ such that $k \geq 3$, which is a subdivision of the house. This yields a contradiction, and by symmetry, we may assume that $H_A \in \mathcal{M}$.

In the rest of the proof, we will consider the graph H' obtained from H by contracting the edge ab. Denoting by u the contracted vertex, observe that the set $S = \{u, c, d\}$ is an independent set in H'. Furthermore, A and B are connected components of $H' - S$ and each vertex of S is adjacent in H' to some vertex in each of A and B. Hence, by Lemma 4.5.17, there exist induced subgraphs H'_A and H'_B of $H'[N[A]]$ and $H'[N[B]]$, respectively, such that $H'_A, H'_B \in \mathcal{S} \cup \mathcal{T} \cup \mathcal{M}$ and S is exactly the sets of extremities of H'_A and H'_B.

By analyzing various cases regarding the structure of these two graphs, we next show that H' contains an induced subdivision of the house. Since the graph H is obtained from the graph H' by subdividing an edge, this will imply that H, and hence also G, contains an induced subdivision of the house, a contradiction.

Note first that, since the graph $H'[N[A]]$ is isomorphic to the graph $H[N[A]]$, it follows that $H'_A \in \mathcal{M}$. We may assume that u is the center in H'_A. The graph H'_B either belongs to $\mathcal{S}$ or to $\mathcal{M}$. Let x be the neighbor of u in the graph H'_A closest to c and let x' be a neighbor of u in the same graph such that u has no neighbors in the interior of the path in A from x to x'. Let P be a c, x'-path in the graph $H'_A - u$. We consider three cases.

Assume first that $H'_B \in \mathcal{S}$. Then the path P and the shortest c, u-path in the graph H'_B, together with the edges ux, ux' form a subdivision of the house in the graph H', a contradiction.

Consider now the case when $H'_B \in \mathcal{M}$ and u is a center of H'_B. Let y be the neighbor of u in the graph H'_B closest to c. Then the path P and the c, y-path in the graph H'_B together with the edges ux, ux', uy form a subdivision of the house in the graph H', a contradiction.

Finally, assume that $H'_B \in \mathcal{M}$ and c is the center of H'_B. Let y be the neighbor of c in the path $H'_B - c$ closest to u. Then, the path P and the u, y-path in the graph H'_B together with the edges ux, ux', cy form a subdivision of the house in the graph H', a contradiction. □

Theorem 4.5.20. *The class of graphs that do not contain an induced subdivision of the house is tame.*

Proof. Let $\mathcal{G}$ be a class of graphs that do not contain an induced subdivision of the house and let G be a graph in $\mathcal{G}$. By Lemma 4.5.18, G is 3-creature-free. By Lemma 4.5.19, G is $\Gamma_{2,2,3}$-induced-minor-free. Since the graph $\Gamma_{2,2,3}$ is an induced minor of the graph $\Gamma_{2,4,4}$, which is isomorphic to the 3-skinny ladder, we conclude that G does not contain a 3-skinny ladder as an induced minor. By Theorem 4.5.13, $\mathcal{G}$ is tame. □

We state the following corollary for later use.

Corollary 4.5.21. *The class of house-induced-minor-free graphs is tame.*

Proof. Immediate from Theorem 4.5.20 and Observation 4.1.2, since the class of house-induced-minor-free graphs is a subclass of the class of graphs that do not contain any induced subdivision of the house. □

4.5.3 Dichotomy results

In this section we prove Theorems 4.5.6 and 4.5.7, characterizing tame graph classes among graph classes excluding a single graph as an induced minor or as an induced topological minor. Recall that a short prism is a graph consisting of two cliques of the same cardinality, and a matching between them and a short theta is a graph consisting of non-adjacent vertices a and b and at least three internally disjoint (anticomplete) paths of length three between them. We restate here two of the results presented in Section 4.3, as we will refer to them in the rest of this section.

Observation 4.5.22. *(1) The class of all short prisms is feral.*

(2) The class of all short thetas is feral.

We proceed with a lemma characterizing graphs that are simultaneously induced minors of both some short prism and some short theta. In order to do that, we need a preparatory technical lemma regarding graph parameters that, informally speaking, measure the distance to minor-closed classes. Let $\mathcal{F}$ be a family of graphs. For a graph G, we define the following parameter: $c_{\mathcal{F}}(G) = \min\{|S|\colon G - S \text{ is } \mathcal{F}\text{-minor-free}\}$ (see, e.g., [96]).

Lemma 4.5.23. *Let $\mathcal{F}$ be a family of graphs and let G and H be graphs such that H is a minor of G. Then, $c_{\mathcal{F}}(H) \leq c_{\mathcal{F}}(G)$.*

Proof. Let $k = c_{\mathcal{F}}(G)$. It suffices to show that any graph G' obtained from G by a single vertex deletion, edge deletion, or edge contraction satisfies $c_{\mathcal{F}}(G') \leq k$. Let $S \subseteq V(G)$ be a set of size k such that $G - S$ is $\mathcal{F}$-minor-free.

If $G' = G - v$ for some vertex $v \in V(G)$, then $S' = S \cap V(G')$ is a set of size at most k such that $G' - S'$ is an induced subgraph of $G - S$ and, hence, $\mathcal{F}$-minor-free.

If $G' = G - e$ for some edge $e \in E(G)$, then $G' - S$ is a subgraph of $G - S$ and, hence, $\mathcal{F}$-minor-free.

Finally, assume that $G' = G/e$ for some edge $e = uv \in E(G)$, and let w be the new vertex. If at least one of u and v belongs to S, then $S' = (S \setminus \{u, v\}) \cup \{w\}$ is a set of size at most k such that $G' - S'$ is an induced subgraph of $G - S$ and, hence, $\mathcal{F}$-minor-free. Otherwise, $u, v \in V(G) \setminus S$, in which case $G' - S$ is a minor of $G - S$ and, hence, $\mathcal{F}$-minor-free. □

We need the following special case of Lemma 4.5.23. A *feedback vertex set* in a graph G is a set $S \subseteq V(G)$ such that $G - S$ is acyclic. The *feedback vertex set number* of G, denoted by $\mathsf{fvs}(G)$, is the minimum cardinality of a feedback vertex set. Since a graph is acyclic if and only if it does not contain the cycle C_3 as a minor, we have $\mathsf{fvs}(G) = c_{\mathcal{F}}(G)$ for $\mathcal{F} = \{C_3\}$; hence, the following holds.

Corollary 4.5.24. *If a graph H is a minor of a graph G, then $\mathsf{fvs}(H) \leq \mathsf{fvs}(G)$.*

This result has the following consequence.

Corollary 4.5.25. *Let H be a graph that is a minor of some short theta. Then, $\mathsf{fvs}(H) \leq 1$.*

Proof. Let k be a positive integer, let G be the k-theta and let H be a minor of G. Since deleting a vertex of degree k from G results in an acyclic graph, we have $\mathsf{fvs}(G) \leq 1$. Hence, Corollary 4.5.24 implies that $\mathsf{fvs}(H) \leq \mathsf{fvs}(G) \leq 1$. □

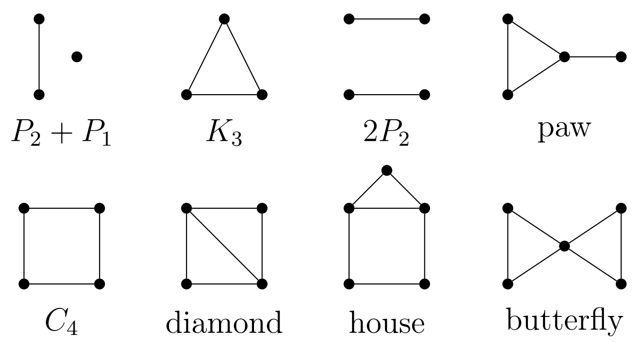


Figure 4.11: Some small graphs.

We can now prove the announced result, a characterization of graphs that are induced minors of some short prism, as well as of some short theta. It turns out that there are only finitely many such graphs. Some of them are depicted in Fig. 4.11.

Lemma 4.5.26. *Let H be a graph that is an induced minor of some short prism and of some short theta. Then H is an induced subgraph of the diamond, the butterfly, or the house.*

Proof. Let $k \geq 3$ and $\ell \geq 3$ be integers and let H be a graph that is an induced minor of the k-prism and of the ℓ-theta. Let G_1 be the k-prism and let G_2 be the ℓ-theta. We prove a few claims about the structure of H.

Claim 1: H is K_4-minor-free.

Proof of Claim 1. Since $\mathsf{fvs}(K_4) = 2$ while $\mathsf{fvs}(G_2) \leq 1$, Corollary 4.5.25 implies that G_2 is K_4-minor-free. But then so is H. ♦

Claim 2: H is $(K_3 + P_1)$-free.

Proof of Claim 2. It suffices to show that G_2 is $(K_3 + P_1)$-induced-minor-free. Suppose for a contradiction that $\widetilde{H} = K_3 + P_1$ is an induced minor of G_2. Label the vertices of $\widetilde{H}$ by p, q, r, s so that s is the isolated vertex and the vertices p, q, r form a triangle. Let $\{X_p, X_q, X_r, X_s\}$ be an induced minor model of $\widetilde{H}$ in G_2. Any two of the sets X_p, X_q, and X_r are connected by an edge in G; using one such edge per pair and, for each of the sets X_p, X_q, and X_r, a path connecting the endpoints of those edges belonging to the set, we obtain a cycle C in G_2. Since every vertex of G_2 either belongs to C or has a neighbor on it, the set X_s cannot be anticomplete to $X_p \cup X_q \cup X_r$; a contradiction with the fact that $\{X_p, X_q, X_r, X_s\}$ is an induced minor model of $\widetilde{H}$ in G_2. ♦

Claim 3: H is gem-free.

Proof of Claim 3. It suffices to show that G_2 is gem-induced-minor-free. Suppose for a contradiction that the gem $\widetilde{H}$ is an induced minor of G_2. Label the vertices of $\widetilde{H}$ by v_1, v_2, v_3, v_4, v_5 so that (v_1, v_2, v_3, v_4) is an induced P_4 and v_5 is universal in $\widetilde{H}$. Let $\{X_v\}_{v \in V(\widetilde{H})}$ be an induced minor model of $\widetilde{H}$ in G_2. Note that G_2 contains two vertices a and b that belong to every cycle. We claim that $\{a, b\} \subseteq X_{v_5}$. This is true, for if up to symmetry $a \notin X_{v_5}$, then there exists some $i \in \{1, 2, 3\}$ such that $a \notin X_{v_i} \cup X_{v_{i+1}} \cup X_{v_5}$ and, hence, the subgraph of G_2 induced by $X_{v_i} \cup X_{v_{i+1}} \cup X_{v_5}$ contains a cycle not containing a, a contradiction. Since $\{a, b\} \subseteq X_{v_5}$, the subgraph of G_2 induced by $\bigcup_{i=1}^{4} X_{v_i}$ is an induced subgraph of $G_2 - \{a, b\}$, that is, of kP_2. But there is a path of length at least 3 in $\bigcup_{i=1}^{4} X_{v_i}$, which cannot be contained in kP_2, a contradiction. ♦

We now use Claims 1–3 to analyze the structure of H, showing that H must be isomorphic to an induced subgraph of the diamond, the butterfly, or the house. Note that any such graph is either a path P_n or an edgeless graph nP_1 for some $n \leq 4$, or one of the graphs depicted in Fig. 4.11.

Since G_1 is cobipartite and H can be obtained by contracting edges of an induced subgraph of G_1, we infer that H is also cobipartite. Then the vertex set of H is a union of two cliques A and B. Since H is K_4-free, each of A and B has at most 3 vertices. Furthermore, By Corollary 4.5.24 it follows that $\mathsf{fvs}(H) \leq 1$ and so at most one of the cliques A and B contains a cycle. Hence, H has at most five vertices. We may assume without loss of generality that $|A| \geq |B|$.

If A has a single vertex, then H is isomorphic to either P_1, $2P_1$, or P_2. If A has two vertices and B has at most one vertex, then H is isomorphic to either P_2, P_3, or K_3. If both A and B have two vertices, let $A = \{a_1, a_2\}$ and $B = \{b_1, b_2\}$. Since H is K_4-minor-free, A and B are not complete to each other. Up to symmetry, let $a_1b_1 \notin E(H)$. Depending on the existence of other edges between A and B we get that H is isomorphic to either $2P_2$, P_4, C_4, the paw, or the diamond.

Consider now the case when $A = \{a_1, a_2, a_3\}$. If B is empty, then $H \cong K_3$. If B consists of a single vertex, then this vertex has either one or two neighbors in A, since H is $\{K_4, K_3 + P_1\}$-free. In the first case, it follows that H is a paw. In the second case H is a diamond. Finally, assume that B consists of two vertices b_1 and b_2. If b_1 and b_2 have a common neighbor in A, we may assume that they are both adjacent to the vertex a_1. If H has no other edges, then H is a butterfly. If, up to symmetry, $a_2b_1 \in E(H)$, then $a_2b_2 \notin E(H)$ and $a_3b_1 \notin E(H)$ since H is K_4-free. Since H is not isomorphic to the gem, we have that $a_3b_2 \in E(H)$, and, hence, $\mathsf{fvs}(H) = 2$, which is in contradiction with Corollary 4.5.25.

If b_1 and b_2 have no common neighbors in A, up to symmetry we may assume that $a_1b_1 \in E(H)$ and $a_2b_2 \in E(H)$, while a_1b_2 and a_2b_1 are non-edges in H. Since H is K_4-minor-free, neither of the two vertices in B is adjacent to a_3, hence, H is a house. □

We now have everything ready to prove Theorems 4.5.6 and 4.5.7, which we restate for the convenience of the reader.

Theorem 4.5.6. *Let H be a graph and let $\mathcal{G}$ be the class of graphs that do not contain H as an induced minor. Then, the following statements are equivalent:*

(1) $\mathcal{G}$ is tame.

(2) $\mathcal{G}$ is not feral.

(3) H is an induced subgraph of the diamond, the butterfly, or the house.

(4) H is an induced minor of the butterfly or of the house.

Proof. Clearly, (1) implies (2). We next prove that (2) implies (3). Assume that $\mathcal{G}$ is not feral. By Observation 4.5.22, neither the class of short prisms nor the class of short thetas is a subclass of $\mathcal{G}$. Hence, there exist integers $k \geq 3$, $\ell \geq 3$ such that H is an induced minor of the k-prism and of the ℓ-theta. By Lemma 4.5.26, it follows that H is an induced subgraph of the diamond, the butterfly, or the house.

Since the diamond is an induced minor of the house, (3) implies (4).

It remains to prove that (4) implies (1). If H is an induced minor of the butterfly, then $\mathcal{G}$ is a subclass of the class of butterfly-induced-minor-free graphs and $\mathcal{G}$ is tame by Theorem 4.5.16. If H is an induced minor of the house, then $\mathcal{G}$ is a subclass of the class of the class of house-induced-minor-free graphs and $\mathcal{G}$ is tame by Corollary 4.5.21. □

Theorem 4.5.7. *Let H be a graph and let $\mathcal{G}$ be the class of graphs that do not contain an induced subdivision of H. Then, the following statements are equivalent:*

(1) $\mathcal{G}$ is tame.

(2) $\mathcal{G}$ is not feral.

(3) H is an induced subgraph of $2P_2$, the diamond, or the house.

(4) H is an induced topological minor of $2P_2$ or of the house.

Proof. Clearly, (1) implies (2). We next show that (2) implies (3). Let $\mathcal{G}$ be non-feral. By Observation 4.5.22, neither the class of short prisms nor the class of short thetas is a subclass of $\mathcal{G}$. Hence, there exist integers $k \geq 3$, $\ell \geq 3$ such that the k-prism G_1 and the ℓ-theta G_2 both contain an induced subdivision of H. In other words, H is an induced topological minor of both G_1 and G_2, and in particular, an induced minor of both G_1 and G_2. It follows from Lemma 4.5.26 that H is an induced subgraph of the diamond, the butterfly, or the house. Note that every proper induced subgraph of the butterfly is also an induced subgraph of either $2P_2$, the diamond, or the house (see Fig. 4.11). Hence, to complete the proof of the implication, it suffices to show that H is not a butterfly.

Suppose for a contradiction that H is the butterfly. Then, G_2 contains an induced subdivision of the butterfly. Let F be an induced subdivision of the butterfly contained in G_2 and let x be the vertex of degree 4 in F (all other vertices in F are of degree 2). Let a and b be the two vertices of degree ℓ in G_2. Every induced cycle in G_2 (and thus in F) contains both a and b, so any two induced cycles in G_2 have at least two common vertices. But the two cycles of F intersect only in x, a contradiction.

Since the diamond is an induced topological minor of the house, (3) implies (4).

It remains to prove that (4) implies (1). If H is an induced topological minor of $2P_2$, then $\mathcal{G}$ is a subclass of the class of $2P_2$-free graphs. The class of $2P_2$-free graphs is tame (by Corollary 4.1.6), and we infer that the class $\mathcal{G}$ is tame as well. If H is an induced topological minor of the house, then the class $\mathcal{G}$ is a subclass of the class of graphs that do not contain any induced subdivision of the house, and $\mathcal{G}$ is tame by Theorem 4.5.20. □

4.6 Recognition algorithms

In this section we give polynomial-time algorithms for the recognition of maximal tame graph classes appearing in our dichotomy theorems Theorems 4.5.6 and 4.5.7 (except for the class of $2P_2$-free graphs, which we already discussed). We first prove Theorem 4.5.9, dealing with graphs containing the house as an induced topological minor, in Section 4.6.1. The result for graphs containing the house as an induced minor and developed in Section 4.6.2 by a reduction to the induced topological minor case. Finally, we explain in Section 4.6.3 a result due to Dumas and Hilaire (personal communication, 2024) leading to a polynomial-time recognition algorithm for determining if a given graph contains the butterfly as an induced minor.

Note that by n and m we denote the number or vertices and edges of input graph, respectively. When discussing the running times of algorithms, we will slightly abuse the notation by writing $\mathcal{O}(m)$ instead of $\mathcal{O}(n+m)$ for linear running time.

4.6.1 Proof of Theorem 4.5.9

In order to recognize graphs excluding the house as an induced topological minor, we first characterize this graph class in terms of two families of forbidden induced subgraphs. Then, we apply the three-in-a-tree algorithm by Chudnovsky and Seymour [64] and a result due to Trotignon and Pham [228].

Building on the terminology of Trotignon and Pham [228], we say that a *long unichord* in a graph is an edge that is the unique chord of some cycle of length at least 5. A graph is *long-unichord-free* if it does not contain any long unichord. A *long theta* is any theta graph other than $K_{2,3}$, that is, a graph of the form $\Gamma_{i,j,k}$ with $\min\{i,j,k\} \geq 2$ and $\max\{i,j,k\} \geq 3$. A graph is *long-theta-free* if it does not contain any long theta as an induced subgraph. These concepts lead to the following characterization of the class of graphs not containing any induced subdivision of the house.

Lemma 4.6.1. *A graph G does not contain any induced subdivision of the house if and only if G is long-unichord-free and long-theta-free.*

Proof. Note that if a graph G contains a long unichord ab, then G contains a cycle C with length at least 5 such that ab is the unique chord of C. In this case, the subgraph of G induced by $V(C)$ is isomorphic to a graph of the form $\Gamma_{1,j,k}$ for some two integers $j \geq 2$ and $k \geq 3$. The converse also holds, hence, G contains a long unichord if and only if G contains an induced subgraph isomorphic to a graph of the form $\Gamma_{1,j,k}$ with $j \geq 2$ and $k \geq 3$. Similarly, G contains an induced long theta if and only if it contains an induced subgraph isomorphic to a graph of the form $\Gamma_{i,j,k}$ such that $i, j \geq 2$ and $k \geq 3$.

Since the house is isomorphic to the graph $\Gamma_{1,2,3}$, any subdivision of the house is isomorphic to the graph $\Gamma_{i,j,k}$ for some $i \geq 1$, $j \geq 2$, and $k \geq 3$. Hence, a graph G contains an induced subdivision of the house if and only if G contains an induced subgraph isomorphic to the graph $\Gamma_{i,j,k}$ for some $i \geq 1$, $j \geq 2$, $k \geq 3$. As shown above, the cases $i = 1$ and $i \geq 2$ correspond to the cases when G contains a long unichord and an induced long theta, respectively. □

By Lemma 4.6.1, we can determine whether a given graph contains an induced subdivision of the house by testing if it has a long unichord or an induced long theta. The former problem has already been solved in the literature, as follows.

Theorem 4.6.2 (Trotignon and Pham [228])**.** *Deciding whether a given graph G has a long unichord can be performed in time $\mathcal{O}(n^4m^2)$.*

In order to test for an induced long theta, we modify an algorithm by Chudnovsky and Seymour [64] to determine whether a given graph G contains an induced theta. The key ingredient in the proof is an efficient solution to the *three-in-a-tree problem*, which takes as input a graph G and a set $X \subseteq V(G)$ with $|X| = 3$, and the task is to determine whether G contains an induced subgraph T such that T is a tree and $X \subseteq V(T)$. Chudnovsky and Seymour gave an algorithm for the three-in-a-tree problem running in time $\mathcal{O}(mn^2)$ (see [64]). This time complexity was significantly improved by Lai et al. [160], who gave an algorithm running in time $\mathcal{O}(m \log^2 n)$.

Proposition 4.6.3. *Deciding whether a given graph G contains an induced long theta can be performed in time $\mathcal{O}(n^8 m \log^2 n)$.*

Proof. Let G be a graph with n vertices and m edges. Enumerate all five-tuples (a, b, v_1, v_2, v_3) of distinct vertices such that vertex a is adjacent to each of the vertices in $\{b, v_1, v_2\}$, vertex b is adjacent to v_3, vertices in $\{v_1, v_2, v_3\}$ are pairwise non-adjacent, and vertices in $\{b, v_1, v_2\}$ are pairwise non-adjacent. For each such five-tuple (a, b, v_1, v_2, v_3), enumerate all subsets $X \subseteq V(G)$ such that a and b have no neighbors in X, and v_1, v_2, v_3 each have exactly one neighbor in X, and each member of X is adjacent to at least one of v_1, v_2, v_3 (it follows that $|X| \leq 3$). For each such choice of X, let G' be obtained from G by deleting a, b and all vertices adjacent to one of a, b, v_1, v_2, v_3 except for the members of $\{v_1, v_2, v_3\} \cup X$ and test whether there is an induced tree in G' containing all of v_1, v_2, v_3. Indeed, we claim that G contains an induced long theta if and only if there is some choice of a, b, v_1, v_2, v_3, and X such that G' contains an induced subgraph T such that T is a tree and $\{v_1, v_2, v_3\} \subseteq V(T)$. Assuming the claim, we have to run the three-in-a-tree algorithm at most n^8 times, and each one takes time $\mathcal{O}(m \log^2 n)$.

It remains to show the claim. Suppose first that G contains an induced long theta, that is, an induced subgraph H isomorphic to a graph of the form $\Gamma_{i,j,k}$ such that $i, j \geq 2$ and $k \geq 3$. Let P^1, P^2, and P^3 be the three edge-disjoint paths forming $\Gamma_{i,j,k}$ of lengths i, j, and k, respectively. Let a be a vertex of degree 3 in H, let v_1 and v_2 be the neighbors of a in P^1 and P^2, respectively, let b be the neighbor of a in P^3, and let v_3 be the neighbor of b in P^3 other than a. For $i \in \{1, 2, 3\}$, let x_i be the neighbor of v_i in P^i such that $X = \{x_1, x_2, x_3\}$ is disjoint from $\{a, b\}$. Let G' be the graph obtained from G by deleting a, b and all vertices adjacent to one of a, b, v_1, v_2, v_3 except for the members of $\{v_1, v_2, v_3\} \cup X$. Then $T = H - \{a, b\}$ is an induced subgraph of G' such that T is a tree and $\{v_1, v_2, v_3\} \subseteq V(T)$.

Conversely, suppose that there is some choice of a, b, v_1, v_2, v_3, and X in G such that G' contains an induced subgraph T such that T is a tree and $\{v_1, v_2, v_3\} \subseteq V(T)$. We may assume that T is an inclusion-minimal induced subtree of G' such that $\{v_1, v_2, v_3\} \subseteq V(T)$. By the minimality of T, the tree T has at most three leaves, and if it has three leaves, then the leaves are exactly the vertices v_1, v_2, and v_3. If T has only two leaves, then T is a path from v_p to v_q having v_r as an internal vertex, where $\{p, q, r\} = \{1, 2, 3\}$. However, this means that v_r two neighbors in X, a contradiction. Hence, T has exactly three leaves, namely v_1, v_2, and v_3, and, furthermore, the minimality of T implies that T is isomorphic to a subdivision of a claw. It follows that the subgraph of G induced by $V(T) \cup \{a, b\}$ is a theta; in fact, it is a long theta, since the path between the two vertices of degree three containing b has length at least three. □

Theorem 4.5.9. *Determining if a given graph G contains the house as an induced topological minor can be done in time $\mathcal{O}(n^8 m \log^2 n)$.*

Proof. The result follows from Lemma 4.6.1, Proposition 4.6.3, and Theorem 4.6.2. □

4.6.2 Proof of Theorem 4.5.8

In this section we show that detecting if a graph contains the house as an induced minor can be done in polynomial time. To reduce this problem to the induced topological minor case, we show (in Theorem 4.6.5) that the existence of the house as the induced minor in a graph G is equivalent to the existence of the house as an induced topological minor or an induced long twin wheel in G, that is, a graph obtained from a cycle of length at least five by replacing a vertex with a pair of adjacent vertices with the same closed neighborhoods.

First we show a useful result about graphs that do not contain the house as an induced topological minor (or, equivalently, graphs that do not contain any induced subdivision of the house). Given a graph G, a subgraph H of G, and a vertex $v \in V(G) \setminus V(H)$, we say that v is a *pendant* of H if v is adjacent to a single vertex of H.

Lemma 4.6.4. *Let G be a graph that does not contain any induced subdivision of the house, H be a hole in G and $v \in V(G) \setminus V(H)$ be a vertex with a neighbor in H. Then one of the following is true: v is pendant of H, or the neighbors of v in H are exactly three vertices that are consecutive, or v is universal for H.*

Proof. Let G, H, v be as stated and let $v_1, \dots, v_k$ be vertices of H in a cyclic order. If v has only one neighbor in H, or is universal for H, we are done, so we may assume that v has at least two neighbors in H and at least one non-neighbor in H. If v has exactly two neighbors in H, then $V(H) \cup \{v\}$ induce a subdivision of the house in G. It follows that v has at least three neighbors in H. Without loss of generality we may assume that $vv_1 \in E(G)$ and $vv_k \notin E(G)$. Let v_i and v_j be neighbors of v in H such that $1 < i < j < k$ and no vertex v_ℓ with $i < \ell < j$ is adjacent to v. If v_1, v_i, v_j are not consecutive, then v and the vertices of the v_1, v_j-path in $H - v_k$ induce a subdivision of the house in G; a contradiction. Hence, v_1, v_i, v_j are consecutive in H, that is, $i = 2$ and $j = 3$. If there is a neighbor of v in G among the vertices $\{v_4, \dots, v_{k-1}\}$, let v_ℓ be the one with the largest index. Then v and the vertices of the v_2, v_ℓ-path in $H - v_3$ induce a subdivision of the house in G; a contradiction. It follows that such a vertex v_ℓ cannot exist and v has exactly three consecutive neighbors in H. □

A *long twin wheel* is a graph consisting of a hole H of length at least 5 and of another vertex v, called the *center*, such that v has degree three and the neighborhood of the center induces a connected graph in H.

Theorem 4.6.5. *Let G be a graph. Then the following statements are equivalent.*

1. *G does not contain the house as an induced minor.*

2. *G does not contain an induced subdivision of the house and does not contain an induced long twin wheel.*

Proof. First we prove that (1) implies (2). Let G be a graph that does not contain the house as an induced minor. Since an induced topological minor is a particular case of an induced minor, it follows that G does not contain an induced subdivision of the house. Suppose for a contradiction that G contains a long twin wheel H as an induced subgraph. Let $v_0, v_1, \dots, v_k$ be vertices of the hole in H, in a cyclic order, and let v be the center of H. Up to symmetry, we may assume that v_0, v_1, v_2 are the neighbors of v in H. Contracting all the edges of the hole in H, except those incident with v_k and v_2 gives a house in G, a contradiction with the assumption that G is house-induced-minor-free.

Now we prove that (2) implies (1). Let G be a graph that does not contain an induced subdivision of the house and does not contain an induced long twin wheel. Let H be a house obtained from the 5-cycle with vertices p, q, r, s, t in cyclic order by adding to it the chord qt and assume that G contains H as an induced minor. Let $\{X_v\}_{v\in V(H)}$ be an induced minor model of H in G. By Proposition 4.5.12, we may assume that the sets X_s, X_r and X_p are all singletons and let $X_s = \{a\}$, $X_r = \{b\}$ and $X_p = \{c\}$. For simplicity, we denote sets X_t and X_q by A and B, respectively. Then G contains two connected subsets A and B and vertices $a, b, c \in V(G) \setminus (A \cup B)$ such that a and b are adjacent, a has a neighbor in A but not in B, b has a neighbor in B but not in A, c has a neighbor in both A and B but is not adjacent to any of a and b, and there is an edge between A and B.

By assumption, there exists an edge in G with endpoints in A and B, so the subgraph of G obtained from $G[A\cup B\cup\{a, b\}]$ by deletion of the edge ab is connected. Hence, there is an induced a, b-path in $G[A \cup B \cup \{a, b\}] - ab$. Note that every such path is of length at least 3. Let $\mathcal{P}$ be the set of all such induced a, b-paths in $G[A\cup B\cup \{a, b\}]-ab$. Given $P \in \mathcal{P}$, let H^P be the hole consisting of the edge ab and of the path P, let P^A be a shortest path from c to the hole H^P such that all vertices of P^A other than c belong to A, and let P^B be a shortest path from c to the hole H^P such that all vertices of P^B other than c belong to B. Finally, let $P \in \mathcal{P}$ be a path that minimizes the number $|E(H^P)| + |E(P^A)| + |E(P^B)|$. In the rest of the proof we simply denote by H the hole H^P.

Let c_A and c_B be vertices on P^A and P^B, respectively, that have neighbors on H.

Assume first that both P^A and P^B are of length one. Then $c = c_A = c_B$ has at least two neighbors in H. As G is a graph without an induced subdivision of the house, by Lemma 4.6.4 it follows that c is either universal for H, or has exactly three consecutive neighbors in H. However, since c is not adjacent to a and b, it follows that c has exactly three consecutive neighbors in H. Thus, H has at least five vertices and together with c induces a long twin wheel, a contradiction to the definition of G. Hence, up to symmetry, we may assume that $c \neq c_A$ and thus P_A is of length at least two.

All internal vertices of P_A belong to A, so c_A is not adjacent to b. Then, by Lemma 4.6.4, it follows that c_A either has one neighbor in H or it has exactly three consecutive neighbors in H. Assume first that c_A has exactly three consecutive neighbors in H. If $|V(H)| \geq 5$, then H together with c_A induces a long twin wheel, a contradiction. This implies that H is a hole on four vertices. Let $\{a, b, y, x\}$ be vertices in H, in cyclic order. Then defining H' to be the hole induced by $\{a, b, y, c_A\}$ implies that the corresponding shortest paths from c to H' are defined as $Q^A = P^A - x$ and $Q^B = P^B$. Altogether we have that $|E(H')|+|E(Q^A)|+|E(Q^B)| < |E(H)|+|E(P^A)|+|E(P^B)|$, which contradicts the minimality of P. Hence, c_A has exactly one neighbor on the hole H.

By symmetry, a similar argumentation can be used to verify that c_B (if not equal to c) has exactly one neighbor on H. Moreover, if $c = c_B$, then by Lemma 4.6.4 it follows that c either has three consecutive neighbors on H, or c has exactly one neighbor in H. In the first case it follows that H and c together induce a long twin wheel, a contradiction to definition of G. Hence, in either case, c_B also has exactly one neighbor on the hole H.

If there is some edge uv in G such that u is an internal vertex of the path P_A, $u \neq c_A$, and v belongs to the hole H, then the path consisting of the c, u-subpath of P_A and of the edge uv can be taken instead of the path P_A, contradicting the minimality condition. By symmetry, the same holds for u being the internal vertex of the path P_B, distinct than c_B. It follows that the internal vertices of paths P_A and P_B, distinct than c_A and c_B have no neighbors in H.

If the sets of internal vertices of P_A and P_B are anticomplete to each other, then H together with P_A and P_B induces a subdivision of the house, so we may assume there is some edge connecting the internal vertices of P_A and P_B. Let x be the neighbor of c_A in H, and let y be the neighbor of c_B in H. Note that $x \neq y$, since $x \in A$, $y \in B$. We know that there is an x, y-path in $G[V(P_A) \cup V(P_B)]$. Let Q be a shortest such path. The existence of the edge connecting the internal vertices of P_A and P_B ensures that $c \notin V(Q)$. Let R be the x, y-path in the hole H that contains vertices a and b, and let S be the xy-path in $H - a$.

Clearly, Q is of length at least three, and at least one of the paths R and S is of length at least two, so the vertices of paths R, S, Q induce an $\Gamma_{i,j,k}$ with $i \geq 1$, $j \geq 2$, $k \geq 3$. But then $\Gamma_{i,j,k}$ is an induced subdivision of the house in G, a contradiction. □

Theorem 4.5.8. *Determining if a given graph G contains the house as an induced minor can be done in time $\mathcal{O}(n^8 m \log^2 n)$.*

Proof. Let $G = (V, E)$ be a graph. By Theorem 4.6.5, in order to test if G is house-induced-minor-free, it suffices to check whether G contains an induced subdivision of the house and whether G contains an induced long twin wheel. If any of these conditions is satisfied, then G is not house-induced-minor-free; otherwise, it is. The former condition can be tested in polynomial time by Theorem 4.5.9. The latter condition can also be tested in polynomial time, as follows. It is not difficult to verify that G contains an induced long twin wheel if and only if there exist four vertices a, b, c, d in G such that the subgraph of G induced by $\{a, b, c, d\}$ is isomorphic to the diamond, the vertices a and d are non-adjacent, and there exists an a, d-path in the graph obtained from G by deleting from it the vertices in $(N[b] \cup N[c]) \setminus \{a, d\}$ as well as all common neighbors of a and d. Checking this condition over all the four-tuples of vertices of G can be done in time $\mathcal{O}(n^4(n + m))$. □

4.6.3 Detecting the butterfly as an induced minor

Determining if a given graph G contains the butterfly as an induced minor can also be done in polynomial time. This is an immediate consequence of the following characterization of graphs containing the butterfly as an induced minor.

Proposition 4.6.6 (Maël Dumas and Claire Hilaire, personal communication, 2024)**.** *Let G be a graph. Then, G contains the butterfly as an induced minor if and only if there exists a set $X \subseteq V(G)$ inducing a subgraph isomorphic to $2P_2$ and a connected component C of the graph $G - X$ such that every vertex in X has a neighbor in C.*

For completeness, we include a short proof of Proposition 4.6.6 based on Proposition 4.5.12.

Proof. Assume first there exists a set $X \subseteq V(G)$ inducing a subgraph isomorphic to $2P_2$ and a connected component C of the graph $G - X$ such that every vertex in X has a neighbor in C. Deleting all vertices in $V(G) \setminus (X \cup V(C))$ and contracting all edges within C results in the butterfly, showing that G contains the butterfly as an induced minor.

Conversely, assume that the butterfly is an induced minor of G. Fix a graph H isomorphic to the butterfly, let z the vertex of degree 4 in H and let $U = V(H) \setminus \{z\}$. Then U is the set of vertices of degree 2 in H. By Proposition 4.5.12, there exists an induced minor model $\{X_v\}_{v \in V(H)}$ of H in G such that $|X_u| = 1$ for all $u \in U$. Let $X = \bigcup_{u \in U} X_u$. Then, the subgraph of G induced by X is isomorphic to $2P_2$. Furthermore, since the set X_z induces a connected subgraph of $G - X$, there exists a connected component C of the graph $G - X$ such that $X \subseteq V(C)$. Since every vertex in U is adjacent to z in H, every vertex in X is adjacent in G to a vertex in X_z and hence to a vertex in C. This completes the proof. □

Theorem 4.6.7. *Determining if a given graph G contains the butterfly as an induced minor can be done in time $\mathcal{O}(n^5(n+m))$.*

Proof. Immediate from Proposition 4.6.6. Indeed, given a graph $G = (V, E)$, we check all subsets $X \subseteq V(G)$ with $|X| = 4$. For each such subset inducing a $2K_2$, we compute the connected components C of the graph $G - X$, for each of them we compute the neighborhood of $V(C)$ in G and test if $X \subseteq N(V(C))$. If one such pair (X, C) is found, then G contains the butterfly as an induced minor, otherwise, it does not. □

Chapter 5

Extremal number of minimal separators

Many graph algorithms are based on minimal separators of the input graph, and the number of minimal separators directly influences the running time of such algorithms. Typically, the first step in an algorithm based on minimal separators is a complete enumeration of the family of minimal separators of the graph. The most efficient known algorithm that enumerates all the minimal separators in a graph was developed by Berry et al. [25] and runs in time $\mathcal{O}(n(n+m)s)$ where n, m, and s denote the number of vertices, edges, and minimal separators of the input graph, respectively.[1] Therefore, since the number of minimal separators directly influences the running time of such algorithms, it is important not only to identify graph classes with a polynomially bounded number of minimal separators, but also to develop sharp upper bounds for the number of minimal separators in such graph classes.

For example, n-vertex split graphs have no more than n minimal separators [196] and, more generally, the same is true for n-vertex chordal graphs [212] and $2P_2$-free graphs. For n-vertex cographs and, more generally, n-vertex P_4-sparse graphs, Nikolopoulos and Palios established an upper bound of $2n/3$ on the number of minimal separators [187].

In this chapter we address the extremal question of determining the maximum number of minimal separators in an n-vertex graph from a given class, for a number of interrelated graph classes with at most a linear number of minimal separators: threshold graphs, split graphs, cographs, trivially perfect graphs and their complements, pseudo-split graphs, and $2P_2$-free graphs. These graph classes have been studied in the literature from various points of view (see [46]). They admit a variety of characterizations; in particular, they can all be defined with a small set of forbidden induced subgraphs, which is a subset of the set $\{2P_2, P_4, C_4, C_5\}$. For each of these classes, we establish exact values for the maximum number of minimal separators in an n-vertex graph that belongs to the class.

[1] The authors only proved a running time of $\mathcal{O}(n^3)$ per separator but the actual bound is $\mathcal{O}(n(n+m))$ per separator, see [176].

5.1 Preliminary remarks

Given a graph class $\mathcal{G}$ and a positive integer n, we denote by $f_{\mathcal{G}}(n)$ the maximum number of minimal separators over all n-vertex graphs $G \in \mathcal{G}$ (with $f_{\mathcal{G}}(n) = 0$ if $\mathcal{G}$ contains no n-vertex graph). If $\mathcal{G}$ is the class of all graphs, then $f_{\mathcal{G}}(n)$ is in $\mathcal{O}\left(((1+\sqrt{5})/2)^n \cdot n\right)$ [113,123]. In this chapter we determine the exact values of $f_{\mathcal{G}}(n)$ for various graph classes and all values of n. Our results are summarized in Table 5.1 and presented in detail in Sections 5.2 and 5.3.

Graph class $\mathcal{G}$	Forbidden induced subgraphs	$f_{\mathcal{G}}(n)$
threshold graphs [67]	$\{2P_2, P_4, C_4\}$	$\lceil (n-1)/2 \rceil$
trivially perfect graphs [126]	$\{P_4, C_4\}$	$\lceil (n-1)/2 \rceil$
co-trivially perfect graphs	$\{2P_2, P_4\}$	$\lceil 2n/3 \rceil - 1$
cographs [73]	$\{P_4\}$	$\lceil 2n/3 \rceil - 1$
split graphs [108]	$\{2P_2, C_4, C_5\}$	$n - \lfloor \log n \rfloor - a(n)$
pseudo-split graphs [169]	$\{2P_2, C_4\}$	$n - \lfloor \log n \rfloor + b(n)$
$2P_2$-free graphs	$\{2P_2\}$	n

Table 5.1: Summary of our results. Functions a and b in the right column satisfy $a : \mathbb{N} \to \{0, 1\}$ and $b : \mathbb{N} \to \{-1, 0, 1, 2\}$. Moreover, $\log n = \log_2 n$.

The family of minimal separators of a disconnected graph can be computed from the families of minimal separators of its components, and a similar statement holds for graphs whose complements are disconnected.

Recall that Corollaries 3.2.2 and 3.2.3 describe the number of separators in the graph in terms of the number of minimal separators in its connected components, or in connected components in the complement of the graph. We restate them below to keep this chapter self-contained.

Corollary 3.2.2. *Let G be a disconnected graph, with components $G_1, \ldots, G_k$. Then $s(G) = \sum_{i=1}^{k} s(G_i) + 1$.*

Corollary 3.2.3. *Let $G_1, \ldots, G_k$ be graphs and let G be the join of $G_1, \ldots, G_k$. Then $s(G) = \sum_{i=1}^{k} s(G_i)$.*

An immediate corollary is that given a universal vertex v in graph G we have that $s(G) = s(G - v)$. The following results from Section 3.2 will be useful for proofs in this chapter. They describe the influence of the vertex deletion on the number of minimal separators in the graph and we restate them below.

Proposition 3.2.13. *Let G be a graph having a pair of true twins v, w with $v \neq w$. Then $s(G) = s(G - v)$.*

Proposition 3.2.14. *Let G be a graph having a pair of false twins v, w with $v \neq w$. Then*

$$s(G - v) \leq s(G) \leq s(G - v) + 1\,.$$

Proposition 3.2.15. *Let G be a graph and let v be a simplicial vertex in G. Then*

$$s(G - v) \leq s(G) \leq s(G - v) + 1\,.$$

We begin this study with the result that considers four known tame graph classes, defined as follows.

A *distance-hereditary graph* is a graph G satisfying that the distances in any connected induced subgraph of G are the same as in G. An *interval graph* is a graph whose vertex set can be associated with intervals on the real line in such a way that two vertices are adjacent if and only if the associated intervals have a nonempty intersection. If no interval is properly contained in the other one, a graph is said to be a *proper interval*. Note that every interval graph is chordal.

Theorem 5.1.1. *Let $\mathcal{G}$ be one of the following classes: proper interval graphs, interval graphs, chordal graphs, distance-hereditary graphs. Then for all $n \in \mathbb{N}$, we have*

$$f_{\mathcal{G}}(n) = \begin{cases} n-2, & \text{if } n \geq 3; \\ n-1, & \text{if } n \in \{1,2\}. \end{cases}$$

Proof. Let $g(1) = 0$, $g(2) = 1$, and $g(n) = n-2$ for all $n \in \mathbb{N}$ with $n \geq 3$. We prove that $f_{\mathcal{G}}(n) = g(n)$ by induction on n. Each graph with at most three vertices is a proper interval graph (end hence an interval graph, and a chordal graph) and also a distance-hereditary graph. For $n = 1$, the only graph to consider is P_1, hence $f_{\mathcal{G}}(1) = s(P_1) = 0 = g(1)$ and the statement holds. For $n = 2$, there are two graphs in $\mathcal{G}$, namely P_2 and $2P_1$, hence $f_{\mathcal{G}}(2) = \max\{s(P_2), s(2P_1)\} = 1 = g(2)$ and the statement holds. For $n = 3$, there are four graphs in $\mathcal{G}$, namely $3P_1$, $P_2 + P_1$, P_3, and K_3, hence $f_{\mathcal{G}}(3) = \max\{s(3P_1), s(P_2+P_1), s(P_3), s(K_3)\} = 1 = g(3)$.

Let now $n \geq 4$ and suppose that $f_{\mathcal{G}}(k) = g(k)$ for all $k \in \{1,\ldots,n-1\}$. Since $n \geq 4$, we have $g(n) = n-2$ and $g(n-1) = n-3$. We want to show that $f_{\mathcal{G}}(n) = n-2$. To show the inequality $f_{\mathcal{G}}(n) \geq n-2$, it suffices to consider the n-vertex path, P_n. Since P_n is a proper interval graph (and hence an interval graph and a chordal graph) as well as a distance-hereditary graph, and every internal vertex of a path forms a minimal separator, we have $f_{\mathcal{G}}(n) \geq s(P_n) \geq n-2$. To show the inequality $f_{\mathcal{G}}(n) \leq n-2$, we need to show that $s(G) \leq n-2$ for every n-vertex graph $G \in \mathcal{G}$. We show that G contains a vertex v such that $s(G) \leq s(G-v)+1$. This will suffice, since then we can apply the induction hypothesis to $G - v \in \mathcal{G}$ to infer that $s(G-v) \leq g(n-1) = n-3$ and consequently $s(G) \leq n-2$, as claimed. Suppose first that G is chordal. A theorem of Dirac [95] states that every minimal separator in a chordal graph is a clique. This implies that every chordal graph has a simplicial vertex. Let v be a simplicial vertex in G. Using Proposition 3.2.15, we infer that $s(G) \leq s(G-v)+1$. Suppose now that G is a distance-hereditary graph. By a result of Bandelt and Mulder [14], G has a pair of distinct vertices v and w such that either v, w are a pair of true twins, v, w are a pair of false twins, or $N_G(v) = \{w\}$. Applying Propositions 3.2.13 to 3.2.15, respectively, shows that $s(G) \leq s(G-v)+1$. This completes the proof. □

5.2 Subclasses of Cographs

Recall that *cographs* are defined as graphs that can be constructed starting from copies of the one-vertex graph using the operations of disjoint union and complementation (see, e.g., [47]). Now we consider the subclasses of cographs: the classes of threshold and trivially perfect graphs. A graph $G = (V, E)$ is *threshold* if there exists a vertex weight function $w : V \to \mathbb{R}_+$ and a threshold $t \in \mathbb{R}_+$ such that a set $X \subseteq V$ is independent in G if and only if $\sum_{x \in X} w(x) \leq t$. Threshold graphs were introduced by

Chvátal and Hammer in [67] and characterized as exactly the $\{2P_2, P_4, C_4\}$-free graphs. Chvátal and Hammer also characterized threshold graphs as exactly the graphs that can be built from the one-vertex graph by iteratively adding either an isolated or a universal vertex.

Theorem 5.2.1 (Chvátal and Hammer [67]). *A graph G is threshold if and only if $G \cong P_1$ or G has a vertex v that is either universal or isolated and such that $G - v$ is threshold.*

A graph G is *trivially perfect* if for every induced subgraph H of G the independence number of H equals the number of maximal cliques in H. Trivially perfect graphs were introduced by Golumbic in [126] and characterized as exactly the $\{P_4, C_4\}$-free graphs. Golumbic also proved a composition theorem characterizing trivially perfect graphs as exactly the graphs that can be built from the one-vertex graph by an iterative application of the operations of disjoint union and addition of a universal vertex.

Theorem 5.2.2 (Golumbic [126]). *Let G be a trivially perfect graph. Then either $G \cong P_1$, or G has a universal vertex v such that $G - v$ is trivially perfect graph, or G is disconnected graph every component of which is a trivially perfect graph.*

These structural results lead to the following.

Theorem 5.2.3. *Let $\mathcal{G}$ be the class of threshold graphs or the class of trivially perfect graphs. Then for all $n \in \mathbb{N}$, we have $f_{\mathcal{G}}(n) = \lceil (n-1)/2 \rceil$.*

Proof. Let $g(n) = \lceil (n-1)/2 \rceil$ for all $n \in \mathbb{N}$. Note that g is non-decreasing, that is, $n_1 \leq n_2$ implies $g(n_1) \leq g(n_2)$. Since the class of threshold graphs is contained in the class of trivially perfect graphs, it suffices to show that for all $n \in \mathbb{N}$ we have $f_{\mathcal{G}_1}(n) \leq g(n) \leq f_{\mathcal{G}_2}(n)$, where $\mathcal{G}_1$ is the class of trivially perfect graphs and $\mathcal{G}_2$ is the class of threshold graphs.

Let us first show that $f_{\mathcal{G}_2}(n) \geq g(n)$. We prove the stated inequality by induction on n. For $n = 1$, the only graph to consider is P_1, hence $f_{\mathcal{G}_2}(1) = s(P_1) = 0 = g(1)$ and the statement holds. For $n = 2$, there are two graphs in $\mathcal{G}_2$, P_2 and $2P_1$, hence $f_{\mathcal{G}_2}(2) = \max\{s(P_2), s(2P_1)\} = 1 = g(2)$ and again the statement holds. Let now $n \geq 3$ and suppose that $f_{\mathcal{G}}(k) = g(k)$ for all $k \in \{1, \ldots, n-1\}$. The induction hypothesis implies that $f_{\mathcal{G}_2}(n-2) = g(n-2)$; in particular, there exists an $(n-2)$-vertex threshold graph G' such that $s(G') = g(n-2)$. Let $G = (G' * P_1) + P_1$. By a double application of Theorem 5.2.1, we infer that G is a threshold graph. Furthermore, Corollary 3.2.3 implies that $s(G' * P_1) = s(G')$ and, using the fact that G is disconnected, with one component isomorphic to $G' * P_1$ and one to P_1, Corollary 3.2.2 yields $s(G) = s(G' * P_1) + 1$. It follows that $f_{\mathcal{G}_2}(n) \geq s(G) = s(G') + 1 = g(n-2) + 1 = g(n)$, which is what we wanted to show.

It remains to show that $f_{\mathcal{G}_1}(n) \leq g(n)$. That is, we want to prove that $s(G) \leq g(n)$ for every n-vertex trivially perfect graph.

For $n \in \mathbb{N}$, let $f^1_{\mathcal{G}_1}(n) = \max\{s(G) \mid G \text{ is a connected } n\text{-vertex graph in } \mathcal{G}_1\}$ and $f^0_{\mathcal{G}}(n) = \max\{s(G) \mid G \text{ is a coconnected } n\text{-vertex graph in } \mathcal{G}_1\}$. First, we will develop equalities and inequalities relating $f^0_{\mathcal{G}_1}(n)$, $f^1_{\mathcal{G}_1}(n)$, and $f_{\mathcal{G}_1}(n)$ with values of these functions at smaller arguments. Clearly, we have $f^0_{\mathcal{G}_1}(1) = f^1_{\mathcal{G}_1}(1) = f_{\mathcal{G}_1}(1) = 0$.

For $n > 1$, we claim that

$$\begin{aligned} f^1_{\mathcal{G}_1}(n) &\leq f^0_{\mathcal{G}_1}(n-1)\,, &(5.1)\\ f^0_{\mathcal{G}_1}(n) &\leq \max_{1\leq i\leq n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1\,, &(5.2)\\ f_{\mathcal{G}_1}(n) &= \max\{f^0_{\mathcal{G}_1}(n), f^1_{\mathcal{G}_1}(n)\}\,. &(5.3)\end{aligned}$$

Equation (5.3) follows immediately from the definitions. In the following we prove that $f^1_{\mathcal{G}_1}(n) \leq f^0_{\mathcal{G}_1}(n-1)$. Let G be a connected n-vertex graph from $\mathcal{G}_1$ that satisfies $s(G) = f^1_{\mathcal{G}_1}(n)$. Let U be the set of universal vertices in G and let $k = |U|$. By Theorem 5.2.2, it follows that $k \geq 1$. If $U = V(G)$, then G is complete graph and $f^1_{\mathcal{G}_1}(n) = s(G) = 0$, so the inequality trivially holds. We may thus assume this is not the case. By definition of U, the graph $G - U$ contains no universal vertices, so it follows by Theorem 5.2.2 that the graph $G-U$ has at least two connected components. Let $H_1, H_2, \ldots, H_\ell$, $\ell \geq 2$, be the connected components of $G - U$. Then we have that $s(G) = s(G - U) = \sum_{i=1}^{\ell} s(H_i) + 1$, by Corollaries 3.2.2 and 3.2.3, respectively. Let G' be the disjoint union of graphs $H_1, \ldots, H_\ell, (k-1)P_1$. Then G' is an $(n-1)$-vertex graph belonging to the class $\mathcal{G}_1$. It follows from the definition of G' and Corollary 3.2.2 that $s(G') = \sum_{i=1}^{\ell} s(H_i) + 1$. Finally, we have that $f^1_{\mathcal{G}_1}(n) = s(G) = s(G') \leq f^0_{\mathcal{G}}(n-1)$, implying that $f^1_{\mathcal{G}_1}(n) \leq f^0_{\mathcal{G}_1}(n-1)$, as we wanted to prove.

Next, we prove the inequality $f^0_{\mathcal{G}_1}(n) \leq \max_{1\leq i\leq n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1$. Let G be an n-vertex coconnected graph from $\mathcal{G}_1$ such that $s(G) = f^0_{\mathcal{G}}(n)$ and such that, subject to this equality, the number of components of G is as small as possible. Let k be the number of components of G. Then $k \geq 2$. We claim that $k = 2$. Suppose for a contradiction that $k \geq 3$, and let $G_1, \ldots, G_k$ be the components of G. Since every component G_i, $i \in \{1, \ldots, k\}$ is a connected graph from $\mathcal{G}_1$, each of them contains at least one universal vertex. If every connected component G_i, $i \in \{1, \ldots, k\}$ is complete, then $f^0_{\mathcal{G}_1}(n) = s(G) = 1$ and inequality trivially holds. We may thus assume that there exists one component that is not complete, say G_k. Let U be the set of universal vertices in G_k. Consider the graph G' obtained from G by deleting the vertices in U and adding a vertex set U' such that $|U'| = |U|$, U' is a clique, and all vertices in U' are adjacent to all vertices in $G-U$, except those in G_1. Using Theorem 5.2.2 we infer that $G_k - U$ belongs to $\mathcal{G}_1$ and has no universal vertices. In particular, by Theorem 5.2.2, $G_k - U$ is disconnected. Let $H_1, \ldots, H_\ell$ be connected components of $G_k - U$. Then the components of G' are G_1 and H, where H is the join of the disjoint union of components $G_2, \ldots, G_{k-1}, H_1, \ldots, H_\ell$ with $G'[U']$. Corollary 3.2.2 implies that $s(G') = s(G_1) + s(H) + 1$. From Corollary 3.2.3 we obtain $s(H) = \sum_{i=2}^{k-1} s(G_i) + \sum_{i=1}^{\ell} s(H_i) + 1$ and $s(G_k) = \sum_{i=1}^{\ell} s(H_i) + 1$, so it follows that

$$s(G') = \sum_{i=1}^{k-1} s(G_i) + \sum_{i=1}^{\ell} s(H_i) + 1 + 1 = \sum_{i=1}^{k} s(G_i) + 1 = s(G).$$

Hence G' is an n-vertex coconnected graph that belongs to $\mathcal{G}_1$ and satisfies $s(G) = s(G')$. Since G' has two connected components, we have a contradiction with minimality of k, so $k = 2$. This shows that G has exactly two components, say G_1 and G_2. Let $j = |V(G_1)|$. Then $j \in \{1, \ldots, n-1\}$ and $|V(G_2)| = n - j$. By Corollary 3.2.2 we have $s(G) = s(G_1) + s(G_2) + 1$, so it follows that

$$\max_{1\leq i\leq n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1 \geq f^1_{\mathcal{G}_1}(j) + f^1_{\mathcal{G}_1}(n-j) + 1 \geq s(G_1) + s(G_2) + 1 = s(G).$$

Since $s(G) = f^0_{\mathcal{G}_1}(n)$, the inequality is proved.

We claim that $f^0_{\mathcal{G}_1}(n) \geq f^0_{\mathcal{G}_1}(n-1)$. Let G' be a coconnected $(n-1)$-vertex graph in $\mathcal{G}_1$ that satisfies $s(G') = f^0_{\mathcal{G}_1}(n-1)$. We construct a coconnected n-vertex graph G in $\mathcal{G}_1$ by adding an isolated vertex to G'. Since G' is induced subgraph of G, by Proposition 3.2.10 it follows that $s(G) \geq s(G')$. Thus $f^0_{\mathcal{G}_1}(n) \geq s(G) \geq s(G') = f^0_{\mathcal{G}_1}(n-1)$, as claimed. Since $f^0_{\mathcal{G}_1}(n) \geq f^0_{\mathcal{G}_1}(n-1)$, we infer using inequality (5.1) that $f^0_{\mathcal{G}_1}(n) \geq f^1_{\mathcal{G}_1}(n)$, which, using equality (5.3), implies that $f_{\mathcal{G}_1}(n) = f^0_{\mathcal{G}_1}(n)$. Consequently, inequality (5.1) implies

$$\begin{aligned} f^0_{\mathcal{G}_1}(n) &\leq \max_{1\leq i\leq n-1}(f^0_{\mathcal{G}_1}(i-1) + f^0_{\mathcal{G}_1}(n-i-1)) + 1 \\ &= \max_{1\leq i\leq n-1}(f_{\mathcal{G}_1}(i-1) + f_{\mathcal{G}_1}(n-i-1)) + 1\,, \end{aligned}$$

where we define $f^0_{\mathcal{G}_1}(0) = f_{\mathcal{G}_1}(0) = 0$. In particular, we obtain the following inequality:

$$f_{\mathcal{G}_1}(n) \leq \max_{1\leq i\leq n-1}(f_{\mathcal{G}_1}(i-1) + f_{\mathcal{G}_1}(n-i-1)) + 1, \text{ for all } n > 1\,. \tag{5.4}$$

To complete the proof, we show that the inequality (5.4) leads to the following bound: for all $n \in \mathbb{N}$, $n = 2k$ or $n = 2k+1$, we have $f_{\mathcal{G}_1}(n) \leq k = g(n)$. We prove this using induction on n. For $n = 0$ or $n = 1$, we have that $f_{\mathcal{G}_1}(n) = 0 = k$. For $n = 2$, we have $f_{\mathcal{G}_1}(n) \leq f_{\mathcal{G}_1}(0) + f_{\mathcal{G}_1}(1) + 1 = 1 = k$. Suppose now that $n > 2$ and that the inequality holds for all smaller arguments. We analyze two cases depending on the parity of n.

Suppose first that $n = 2k+2$ for some $k \in \mathbb{N}$. Note that for every $i \in \{1, \ldots, n-1\}$, we have $(i-1)+(n-i-1) = n-2$. Thus, considering inequality (5.4), we analyze the possible ways how to express $n-2 = 2k$ as the sum of two smaller positive integers. Working modulo 2, there are two essentially different ways of how this can be done: either $n-2 = (2i+1) + (2j+1)$ or $n-2 = (2i) + (2j+2)$ for $i, j \in \mathbb{Z}_+$ with $i + j = k - 1$. The induction hypothesis implies that $f_{\mathcal{G}_1}(2i+1) \leq i$, $f_{\mathcal{G}_1}(2j+1) \leq j$, $f_{\mathcal{G}_1}(2i) \leq i$, $f_{\mathcal{G}_1}(2j+2) \leq j+1$. Hence $f_{\mathcal{G}_1}(2i+1) + f_{\mathcal{G}_1}(2j+1) + 1 \leq i+j+1 = k$, $f_{\mathcal{G}_1}(2i) + f_{\mathcal{G}_1}(2j+2) \leq i+j+1+1 = k+1$. Using inequality (5.4), it follows that $f_{\mathcal{G}_1}(n) \leq \max\{k, k+1\} = k+1$, as claimed.

Suppose now that $n = 2k+1$ for some $k \in \mathbb{N}$. Now we have $n-2 = 2k-1 = (2i) + (2j+1)$ for some $i, j \in \mathbb{Z}_+$ with $i + j = k - 1$. The induction hypothesis implies that $f_{\mathcal{G}_1}(2i) \leq i$, $f_{\mathcal{G}_1}(2j+1) \leq j$. Finally, inequality (5.4) implies that $f_{\mathcal{G}_1}(n) \leq f_{\mathcal{G}_1}(2i) + f_{\mathcal{G}_1}(2j+1) \leq i+j+1 = k$, as claimed. This completes the proof. □

Next, we consider the classes of co-trivially perfect graphs and cographs. A graph is *co-trivially perfect* if its complement is trivially perfect. The following characterization of co-trivially perfect graphs is an immediate consequence of Theorem 5.2.2.

Corollary 5.2.4. *Let G be a co-trivially perfect. Then either $G \cong P_1$, or G has an isolated vertex v such that $G - v$ is co-trivially perfect or G is a join of two co-trivially perfect graphs.*

The class of cographs is defined as the smallest class of graphs containing the one-vertex graph that is closed under the operations of disjoint union and join. Cographs were introduced independently several times in the literature and are known to coincide with the P_4-free graphs (see, e.g., [73]). A result of Nikolopoulos and Palios from [187] determines the exact value of $f_{\mathcal{G}}(n)$, where $\mathcal{G}$ is the class of cographs, if n is a power of 2. A more detailed analysis reveals the exact values of $f_{\mathcal{G}}(n)$ for all positive integers n. These values coincide with the values for the class of co-trivially perfect graphs.

Theorem 5.2.5. *Let $\mathcal{G}$ be the class of co-trivially perfect graphs or the class of cographs. Then for all $n \in \mathbb{N}$, we have $f_{\mathcal{G}}(n) = \lceil 2n/3 \rceil - 1$.*

Proof. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the classes of cographs and of trivially perfect graphs, respectively. For $n \in \mathbb{N}$ and $k \in \{1, 2\}$, let $f^1_{\mathcal{G}_k}(n) = \max\{s(G) \mid G$ is a connected n-vertex graph in $\mathcal{G}_k\}$ and $f^0_{\mathcal{G}_k}(n) = \max\{s(G) \mid G$ is a coconnected n-vertex graph in $\mathcal{G}_k\}$. First, we will develop recurrence relations for $f^0_{\mathcal{G}_k}(n)$, $f^1_{\mathcal{G}_k}(n)$, and $f_{\mathcal{G}_k}(n)$. Clearly, we have $f^0_{\mathcal{G}_k}(1) = f^1_{\mathcal{G}_k}(1) = f_{\mathcal{G}_k}(1) = 0$. For $n > 1$ and for $k \in \{1, 2\}$, we claim that

$$f_{\mathcal{G}_k}(n) = \max\{f^0_{\mathcal{G}_k}(n), f^1_{\mathcal{G}_k}(n)\}\,, \tag{5.5}$$
$$f^1_{\mathcal{G}_k}(n) = \max_{1\le i\le n-1}(f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i))\,, \tag{5.6}$$
$$f^0_{\mathcal{G}_k}(n) = \begin{cases} \max\limits_{1\le i\le n-1}(f^1_{\mathcal{G}_k}(i) + f^1_{\mathcal{G}_k}(n-i)) + 1, & \text{if } k = 1\\ f^1_{\mathcal{G}_k}(n-1) + 1, & \text{if } k = 2\,. \end{cases} \tag{5.7}$$

Equation (5.5) follows immediately from the definitions.

We prove the equality (5.6) by proving two inequalities. Let $k \in \{1, 2\}$. The inequality $f^1_{\mathcal{G}_k}(n) \ge \max_{1\le i\le n-1}(f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i))$, is equivalent to showing that $f^1_{\mathcal{G}_k}(n) \ge f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i)$ holds for all $i \in \{1, \ldots, n-1\}$. For each $j \in \{i, n-i\}$, let G^j be a j-vertex graph in $\mathcal{G}_k$ such that $s(G^j) = f_{\mathcal{G}_k}(j)$, and let G be the join of G^i and G^{n-i}. Then G is an n-vertex connected graph in $\mathcal{G}_k$. Moreover, by Corollary 3.2.3, we have $s(G) = s(G^i) + s(G^{n-i}) = f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i)$. This implies $f^1_{\mathcal{G}_k}(n) \ge f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i)$, hence, since i was arbitrary, $f^1_{\mathcal{G}_k}(n) \ge \max_{1\le i\le n-1}(f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i))$.

Let us now establish the converse inequality $f^1_{\mathcal{G}_k}(n) \le \max_{1\le i\le n-1}(f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i))$. Let G be an n-vertex connected graph in $\mathcal{G}_k$ such that $s(G) = f^1_{\mathcal{G}_k}(n)$. Since G is a connected graph in $\mathcal{G}_k$ with more than one vertex, it can be written as the join of two smaller graphs in $\mathcal{G}_k$, say G_1 and G_2. Let $j = |V(G_1)|$. Then $j \in \{1, \ldots, n-1\}$ and $|V(G_2)| = n - j$. By Corollary 3.2.3, we have $s(G) = s(G_1) + s(G_2)$ and consequently,

$$\max_{1\le i\le n-1}(f_{\mathcal{G}_k}(i) + f_{\mathcal{G}_k}(n-i)) \ge f_{\mathcal{G}_k}(j) + f_{\mathcal{G}_k}(n-j) \ge s(G_1) + s(G_2) = s(G).$$

Since $s(G) = f^1_{\mathcal{G}_k}(n)$, the inequality is proved.

In the following we will prove equality (5.7) by proving the two inequalities for each value of k. The inequality $f^0_{\mathcal{G}_1}(n) \ge \max_{1\le i\le n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1$ can be proved by taking G to be the disjoint union of two cographs G^i and G^{n-i} where G^j for $j \in \{i, n-i\}$ is a j-vertex connected cograph such that $s(G^j) = f^1_{\mathcal{G}_1}(j)$. Then G is an n-vertex coconnected cograph and, by Corollary 3.2.2, we have $s(G) = s(G^i) + s(G^{n-i}) = f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i) + 1$. Again, since i was arbitrary, we obtain $f^0_{\mathcal{G}_1}(n) \ge \max_{1\le i\le n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1$. The inequality $f^0_{\mathcal{G}_2}(n) \ge f^1_{\mathcal{G}_2}(n-1) + 1$ can be proved similarly, by taking G to be the disjoint union of graphs G' and P_1, where G' is a connected $(n-1)$-vertex graph in $\mathcal{G}_2$ that satisfies $s(G) = f_{\mathcal{G}_2}(n-1)$. Then G is an n-vertex coconnected graph in $\mathcal{G}_2$ with connected components G' and P_1 and, by Corollary 3.2.2, we have $s(G) = s(G') + s(P_1) + 1 = f^1_{\mathcal{G}_2}(n-1) + 1$. Consequently, we obtain $f^0_{\mathcal{G}_2}(n) \ge s(G) = f^1_{\mathcal{G}_2}(n-1) + 1$.

Next, we prove the inequality $f^0_{\mathcal{G}_1}(n) \le \max_{1\le i\le n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1$. Let G be an n-vertex coconnected cograph such that $s(G) = f^0_{\mathcal{G}_1}(n)$ and such that, subject to this equality, the number of components of G is as small as possible. Let ℓ be the number of components of G. Since G is a coconnected cograph with more than one vertex, G is disconnected, that is, $\ell \geq 2$. We claim that $\ell = 2$. Suppose for a contradiction that $\ell \geq 3$, and let $G_1, \ldots, G_\ell$ be the components of G. Consider the graph G' obtained from G by adding all edges joining a vertex in $G_{\ell-1}$ with a vertex in G_ℓ. Then, the components of G' are $G_1, \ldots, G_{\ell-2}$ and $H_{\ell-1}$ where $H_{\ell-1}$ is the join of $G_{\ell-1}$ and G_ℓ. In particular, it follows that G' is an n-vertex disconnected cograph, hence G' is coconnected. Corollary 3.2.2 implies that $s(G') = \sum_{i=1}^{\ell-2} s(G_i) + s(H_{\ell-1}) + 1$. From Corollary 3.2.3 we obtain $s(H_{\ell-1}) = s(G_{\ell-1}) + s(G_\ell)$, which in turn implies $s(G') = \sum_{i=1}^{\ell} s(G_i) + 1 = s(G) = f^0_{\mathcal{G}_1}(n)$, where the last equality follows again from Corollary 3.2.2. However, since G' has one component less than G, the fact that $s(G') = f^0_{\mathcal{G}_1}(n)$ contradicts the choice of G. This shows that G has exactly two components, say G_1 and G_2. Let $j = |V(G_1)|$. Then $j \in \{1, \ldots, n-1\}$ and $|V(G_2)| = n - j$. Moreover, by Corollary 3.2.2 we have $s(G) = s(G_1) + s(G_2) + 1$. It follows that

$$\max_{1\leq i\leq n-1}(f^1_{\mathcal{G}_1}(i) + f^1_{\mathcal{G}_1}(n-i)) + 1 \geq f^1_{\mathcal{G}_1}(j) + f^1_{\mathcal{G}_1}(n-j) + 1 \geq s(G_1) + s(G_2) + 1 = s(G).$$

As $s(G) = f^0_{\mathcal{G}_1}(n)$, the inequality is proved.

Finally, we prove the inequality $f^0_{\mathcal{G}_2}(n) \leq f^1_{\mathcal{G}_2}(n-1) + 1$. Let G be an n-vertex coconnected graph in $\mathcal{G}_2$ such that $s(G) = f^0_{\mathcal{G}_2}(n)$ and such that, subject to this equality, the number of isolated vertices in G is as small as possible. Let ℓ be the number of isolated vertices in G. By Corollary 5.2.4, G has at least one isolated vertex, that is, $\ell \geq 1$. We claim that $\ell = 1$. Suppose for a contradiction that $\ell \geq 2$, and let I be the set of isolated vertices in G. Let $v \in I$. Consider the graph G' obtained from G by turning the set $I \setminus \{v\}$ into a clique and adding all edges joining vertices in $G - I$ with vertices in $I \setminus \{v\}$. Since any complete graph is in $\mathcal{G}_2$, and $G' - v$ is a join of $K_{\ell-1}$ and $G - I$, we infer that G' is an n-vertex graph in $\mathcal{G}_2$, having exactly one isolated vertex. By Corollaries 3.2.2 and 3.2.3 we have $s(G') = s(G'-v)+1 = s(G-I)+s(K_{\ell-1})+1 = s(G-I)+1 = s(G)$. Since $s(G) = s(G')$, this contradicts the choice of G and shows that G has exactly one isolated vertex, say v. By Corollary 3.2.2 we have $s(G) = s(G-v)+1$. It follows that $f^1_{\mathcal{G}_2}(n-1) + 1 \geq s(G-v) + 1 = s(G) = f^0_{\mathcal{G}_2}(n)$ and the inequality is proved.

To complete the proof, we show that the recurrence relations (5.5)–(5.7) together with initial conditions $f^0_{\mathcal{G}_k}(1) = f^1_{\mathcal{G}_k}(1) = f_{\mathcal{G}_k}(1) = 0$ lead to the following explicit formulas for the function values $f^0_{\mathcal{G}_k}(n)$, $f^1_{\mathcal{G}_k}(n)$, and $f_{\mathcal{G}_k}(n)$. For all $n \in \mathbb{N}$ and for $k \in \{1, 2\}$, we have:

- If $n = 3\ell + 1$ for some $\ell \in \mathbb{Z}_+$, then $f^1_{\mathcal{G}_k}(n) = f^0_{\mathcal{G}_k}(n) = f_{\mathcal{G}_k}(n) = 2\ell$.
- If $n = 3\ell + 2$ for some $\ell \in \mathbb{Z}_+$, then $f^1_{\mathcal{G}_k}(n) = 2\ell$ and $f^0_{\mathcal{G}_k}(n) = f_{\mathcal{G}_k}(n) = 2\ell + 1$.
- If $n = 3\ell + 3$ for some $\ell \in \mathbb{Z}_+$, then $f^1_{\mathcal{G}_k}(n) = f^0_{\mathcal{G}_k}(n) = f_{\mathcal{G}_k}(n) = 2\ell + 1$.

This will indeed suffice, since the statement $f_{\mathcal{G}_k}(n) = \lceil 2n/3 \rceil - 1$ for all $n \in \mathbb{N}$ and for all $k \in \{1, 2\}$ is easily seen to be equivalent to the truthfulness of the above explicit formulas for the values of $f_{\mathcal{G}_k}(n)$. We prove the formulas by induction on n. Let $k \in \{1, 2\}$. For $n = 1$, the formulas state that $f^1_{\mathcal{G}_k}(1) = f^0_{\mathcal{G}_k}(1) = f_{\mathcal{G}_k}(1) = 0$, which is true. For $n = 2$, we have $f^1_{\mathcal{G}_k}(2) = f_{\mathcal{G}_k}(1) + f_{\mathcal{G}_k}(1) = 0$ (by (5.6)), $f^0_{\mathcal{G}_k}(2) = f^1_{\mathcal{G}_k}(1) + f^1_{\mathcal{G}_k}(1) + 1 = 1$ (by (5.7)), and $f_{\mathcal{G}_k}(2) = \max\{f^1_{\mathcal{G}_k}(2), f^0_{\mathcal{G}_k}(2)\} = 1$ (by (5.5)), which coincides with the values given by the explicit formulas. Similarly, for $n = 3$, we have $f^1_{\mathcal{G}_k}(3) = f_{\mathcal{G}_k}(1) + f_{\mathcal{G}_k}(2) = 1$, $f^0_{\mathcal{G}_1}(3) = f^1_{\mathcal{G}_1}(1) + f^1_{\mathcal{G}_1}(2) + 1 = 1$, $f^0_{\mathcal{G}_2}(3) = f^1_{\mathcal{G}_2}(2) + 1 = 1$ and $f_{\mathcal{G}_k}(3) = \max\{f^1_{\mathcal{G}_k}(3), f^0_{\mathcal{G}_k}(3)\} = 1$, which also matches with the explicit formulas. Suppose now that $n > 3$ and that the explicit formulas hold for all smaller arguments. We analyze the three cases depending on the value of $n \bmod 3$.

Suppose first that $n = 3\ell + 1$ for some $\ell \in \mathbb{N}$. Working modulo 3, there are two essentially different ways of how n can be written as the sum of two smaller positive integers: either $n = (3i + 1) + (3j + 3)$ or $n = (3i + 2) + (3j + 2)$ for $i, j \in \mathbb{Z}_+$ with $i + j = \ell - 1$. The induction hypothesis implies that $f_{\mathcal{G}_k}(3i + 1) = f^1_{\mathcal{G}_k}(3i + 1) = 2i$, $f_{\mathcal{G}_k}(3j+3) = f^1_{\mathcal{G}_k}(3j+3) = 2j+1$, $f_{\mathcal{G}_k}(3i+2) = 2i+1$, $f^1_{\mathcal{G}_k}(3i+2) = 2i$, $f_{\mathcal{G}_k}(3j+2) = 2j + 1$, and $f^1_{\mathcal{G}_k}(3j + 2) = 2j$, $f^1_{\mathcal{G}_2}(n - 1) = f^1_{\mathcal{G}_2}(3\ell) = 2\ell - 1$.

Hence $f_{\mathcal{G}_k}(3i + 1) + f_{\mathcal{G}_k}(3j + 3) = f^1_{\mathcal{G}_k}(3i + 1) + f^1_{\mathcal{G}_k}(3j + 3) = 2(i + j) + 1 = 2\ell - 1$, $f_{\mathcal{G}_k}(3i+2)+f_{\mathcal{G}_k}(3j+2) = 2(i+j+1) = 2\ell$, and $f^1_{\mathcal{G}_k}(3i+2)+f^1_{\mathcal{G}_k}(3j+2) = 2(i+j) = 2\ell-2$. Using recurrence relations (5.5)–(5.7), it follows that $f^1_{\mathcal{G}_k}(n) = \max_{1\le q\le n-1}(f_{\mathcal{G}_k}(q) + f_{\mathcal{G}_k}(n - q)) = 2\ell$, $f^0_{\mathcal{G}_1}(n) = \max_{1\le q\le n-1}(f^1_{\mathcal{G}_k}(q) + f^1_{\mathcal{G}_k}(n - q)) + 1 = 2\ell$, and $f^0_{\mathcal{G}_2}(n) = f^1_{\mathcal{G}_k}(n - 1) + 1 = 2\ell - 1 + 1 = 2\ell$, and $f_{\mathcal{G}_k}(n) = 2\ell$, as claimed.

Suppose now that $n = 3\ell + 2$ for some $\ell \in \mathbb{N}$. Now we have either $n = (3i + 1) + (3j + 1)$ for some $i, j \in \mathbb{Z}_+$ with $i + j = \ell$ or $n = (3i' + 3) + (3j' + 2)$ for some $i', j' \in \mathbb{Z}_+$ with $i' + j' = \ell - 1$. The induction hypothesis implies that $f_{\mathcal{G}_k}(3i + 1) = f^1_{\mathcal{G}_k}(3i + 1) = 2i$, $f_{\mathcal{G}_k}(3j + 1) = f^1_{\mathcal{G}_k}(3j + 1) = 2j$, $f_{\mathcal{G}_k}(3i' + 3) = f^1_{\mathcal{G}_k}(3i' + 3) = 2i' + 1$, $f_{\mathcal{G}_k}(3j' + 2) = 2j' + 1$, $f^1_{\mathcal{G}_k}(3j' + 2) = 2j'$, and $f^1_{\mathcal{G}_k}(n - 1) = f^1_{\mathcal{G}_k}(3\ell + 1) = 2\ell$. Hence $f_{\mathcal{G}_k}(3i+1)+f_{\mathcal{G}_k}(3j+1) = f^1_{\mathcal{G}_k}(3i+1)+f^1_{\mathcal{G}_k}(3j+1) = 2(i+j) = 2\ell$, $f_{\mathcal{G}_k}(3i'+3)+f_{\mathcal{G}_k}(3j'+2) = 2(i'+j'+1) = 2\ell$, and $f^1_{\mathcal{G}_k}(3i'+3)+f^1_{\mathcal{G}_k}(3j'+2) = 2(i'+j')+1 = 2\ell-1$. Recurrence relations (5.5)–(5.7) imply that $f^1_{\mathcal{G}_k}(n) = \max_{1\le q\le n-1}(f_{\mathcal{G}_k}(q) + f_{\mathcal{G}_k}(n - q)) = 2k$ and $f^0_{\mathcal{G}_1}(n) = \max_{1\le q\le n-1}(f^1_{\mathcal{G}_k}(q)+f^1_{\mathcal{G}_k}(n-q))+1 = 2\ell+1$, $f^0_{\mathcal{G}_2}(n) = f^1_{\mathcal{G}_k}(n-1)+1 = 2\ell+1$, and $f_{\mathcal{G}_k}(n) = 2\ell + 1$, as claimed.

Finally, let $n = 3\ell + 3$ for some $\ell \in \mathbb{N}$. In this case, we have either $n = (3i + 1) + (3j + 2)$ for some $i, j \in \mathbb{Z}_+$ with $i + j = \ell$ or $n = (3i' + 3) + (3j' + 3)$ for some $i', j' \in \mathbb{Z}_+$ with $i' + j' = \ell - 1$. The induction hypothesis implies that $f_{\mathcal{G}_k}(3i + 1) = f^1_{\mathcal{G}_k}(3i + 1) = 2i$, $f_{\mathcal{G}_k}(3j + 2) = 2j + 1$, $f^1_{\mathcal{G}_k}(3j + 2) = 2j$, $f_{\mathcal{G}_k}(3i' + 3) = f^1_{\mathcal{G}_k}(3i' + 3) = 2i' + 1$, $f_{\mathcal{G}_k}(3j' + 3) = f^1_{\mathcal{G}_k}(3j' + 3) = 2j' + 1$, and $f^1_{\mathcal{G}_k}(n - 1) = f^1_{\mathcal{G}_k}(3\ell + 2) = 2\ell$. Hence $f_{\mathcal{G}_k}(3i+1)+f_{\mathcal{G}_k}(3j+2) = 2(i+j)+1 = 2\ell+1$, $f^1_{\mathcal{G}_k}(3i+1)+f^1_{\mathcal{G}_k}(3j+2) = 2(i+j) = 2\ell$, $f_{\mathcal{G}_k}(3i'+3) + f_{\mathcal{G}_k}(3j'+3) = f^1_{\mathcal{G}_k}(3i'+3) + f^1_{\mathcal{G}_k}(3j'+2) = 2(i' + j' + 1) = 2\ell$. Recurrence relations (5.5)–(5.7) imply that $f^1_{\mathcal{G}_k}(n) = \max_{1\le q\le n-1}(f_{\mathcal{G}_k}(q) + f_{\mathcal{G}_k}(n - q)) = 2\ell + 1$, $f^0_{\mathcal{G}_1}(n) = \max_{1\le q\le n-1}(f^1_{\mathcal{G}_k}(q)+f^1_{\mathcal{G}_k}(n-q))+1 = 2\ell+1$, $f^0_{\mathcal{G}_2}(n) = f^1_{\mathcal{G}_k}(n-1)+1 = 2\ell+1$, and $f_{\mathcal{G}_k}(n) = 2\ell + 1$, as claimed. This completes the proof. □

5.3 Split Graphs, Pseudo-Split Graphs, and $2P_2$-Free Graphs

A *split graph* is a graph that has a *split partition*, that is, a partition (C, I) of its vertex set into a clique C and an independent set I. Split graphs were introduced by Foldes and Hammer in [108] and characterized as exactly the $\{2P_2, C_4, C_5\}$-free graphs. It is not difficult to see that every split graph has a split partition (C, I) such that C is a maximal clique. The following lemma given by Pedrotti and de Mello is stated in [196] without proof. We include a proof for the sake of completeness.

Lemma 5.3.1. *Let G be a split graph with a split partition (C, I) such that C is a maximal clique in G. Then $\mathcal{S}_G = \{N_G(v) \mid v \in I\}$.*

Proof. Suppose first that S is a minimal separator in G. Since G is $2P_2$-free and there are two S-full components in $G - S$, by Lemma 3.1.3 there exists a vertex $v \in V(G)$ such that $S = N_G(v)$. Clearly $G[\{v\}]$ is an S-full component of $G-S$. The graph $G-S$ has an S-full component other than $G[\{v\}]$; let C' be such a component. To complete the proof, we show that there exists some $w \in I$ such that $S = N_G(w)$. Suppose that this is not the case. Then $v \in C$ and $S = (C \setminus \{v\}) \cup (S \cap I)$; in particular, the graph $G-S$ consists of isolated vertices only. Since v is the only vertex in $G-S$ that belongs to C, it follows that $V(C') = \{w'\}$ for some $w' \in I$. Since C' is an S-full component, every vertex in S is adjacent to w', that is, $N_G(v) \subseteq N_G(w')$. By assumption, these two sets are not the same, hence there exists a vertex $z \in N_G(w') \setminus N_G(v)$. However, since $w' \notin S = N_G(v)$, vertices w' and v are non-adjacent. It follows that $z \neq w'$ and hence $z \in C \setminus \{v\}$, which implies $z \in N_G(v)$, a contradiction.

Suppose now that $S = N_G(v)$ for some vertex $v \in I$. Since C is a maximal clique, there exists a vertex $w \in C \setminus S$. Let C' be the component of $G - S$ containing w. Since C is a clique and $S \cup \{w\} \subseteq C$, every vertex in S is adjacent to w. It follows that $G[\{v\}]$ and C' are two distinct S-full components of $G - S$; hence S is a minimal separator. □

Using Lemma 5.3.1, the following extremal result for the class of split graphs can be obtained.

Theorem 5.3.2. *Let $\mathcal{G}$ be the class of split graphs. Then for all $n \in \mathbb{N}$ we have $f_{\mathcal{G}}(n) = n - \min\{k \in \mathbb{N} \mid 2^k + k > n\}$. Consequently, for all $n \in \mathbb{N}$ we have $n - \lfloor \log n \rfloor - 1 \leq f_{\mathcal{G}}(n) \leq n - \lfloor \log n \rfloor$.*

Proof. For $n \in \mathbb{N}$, let $\kappa(n) = \min\{k \in \mathbb{N} \mid 2^k + k > n\}$ and let $g(n) = n - \kappa(n)$. We prove that $f_{\mathcal{G}}(n) = g(n)$ for all $n \in \mathbb{N}$ by proving each of the two inequalities. Let $n \in \mathbb{N}$. First we prove that $f_{\mathcal{G}}(n) \leq g(n)$, that is, that every n-vertex split graph G satisfies $s(G) \leq g(n)$. Fix a split partition (C, I) of G such that C is a maximal clique in G. Let $k = \kappa(n)$ and $\ell = |C|$. Lemma 5.3.1 and the fact that C is a maximal clique imply that $\mathcal{S}_G = \{N_G(x) \mid x \in I\} \subseteq \{X \mid X \subseteq C\} \setminus \{C\}$ and hence $s(G) \leq \min\{|I|, 2^\ell - 1\}$. Suppose for a contradiction that $s(G) > g(n)$. Since $g(n) = n - k$, this implies $n - k + 1 \leq s(G) \leq |I| = n - |C| = n - \ell$, hence $\ell \leq k - 1$. Using the fact that $s(G) \leq 2^\ell - 1$, we infer that $s(G) \leq 2^{k-1} - 1$. On the other hand, the definition of k implies that $2^{k-1} + k - 1 \leq n$, which implies $s(G) \geq n - k + 1 \geq 2^{k-1}$, a contradiction.

We now prove that $f_{\mathcal{G}}(n) \geq g(n)$, that is, that there exists an n-vertex split graph G satisfying $s(G) \geq g(n)$. Let $k = \kappa(n)$. Then $n - k < 2^k$. Let K be a complete graph of order k, let $C = V(K)$, and let $\mathcal{F}$ be a set of subsets of C such that $C \notin \mathcal{F}$ and $|\mathcal{F}| = n - k$. Note that such a set $\mathcal{F}$ exists, since $n - k \leq 2^k - 1 = 2^{|C|} - 1$. Moreover, let $I = \{v_X \mid X \in \mathcal{F}\}$ be a set of $n - k$ new vertices and let G be the split graph with split partition (C, I) obtained from the disjoint union of K and the edgeless graph with vertex set I by adding the edges of the form $\{v_X, w\}$ for all $w \in X \in \mathcal{F}$. By construction, G is an n-vertex split graph with C a maximal clique. Hence, Lemma 5.3.1 implies that $\mathcal{S}_G = \{N_G(v) \mid v \in I\}$. Since distinct vertices in I have distinct neighborhoods, $s(G) = |I| = n - k = g(n)$ and $f_{\mathcal{G}}(n) \geq s(G) = g(n)$, as claimed.

It remains to show that $n - \lfloor \log n \rfloor - 1 \leq g(n) \leq n - \lfloor \log n \rfloor$, or, equivalently, that $\lfloor \log n \rfloor \leq k \leq \lfloor \log n \rfloor + 1$ where $k = \kappa(n)$. The definition of k implies that $2^{k-1} + k - 1 \leq n$. Consequently $k - 1 \leq \log(2^{k-1} + k - 1) \leq \log n$, which implies $k \leq \log n + 1$ and thus, since k is an integer, $k \leq \lfloor \log n \rfloor + 1$. To prove that $k \geq \lfloor \log n \rfloor$, note that if $2^k \geq n$, then $k \geq \log n \geq \lfloor \log n \rfloor$, while if $2^k < n$, then $k = \lfloor \log n \rfloor$ since $k < \log n < k + 1$, where the last inequality follows from $n < 2^k + k < 2^{k+1}$. This completes the proof. $\square$

A graph is *pseudo-split* if it is $\{2P_2, C_4\}$-free. Pseudo-split graphs were characterized independently by Blázsik et al. in [29] and by Maffray and Preissmann in [169] as exactly the graphs that are either split or consist of a split graph G with a split partition (C, I) together with a 5-cycle disjoint from G that is fully adjacent to vertices in C and fully non-adjacent to vertices in I. Given a graph G, a *pseudo-split partition* of G is a partition (C, I, S) of $V(G)$ into a clique C, an independent set I and a set S such that either $S = \emptyset$ or the subgraph of G induced by S is a 5-cycle fully adjacent to vertices in C and fully non-adjacent to vertices in I. A graph G is pseudo-split if and only if it has a pseudo-split partition (see [29, 169]). This structural result together with Theorem 5.3.2 leads to the following extremal result.

Theorem 5.3.3. *Let $\mathcal{G}$ be the class of pseudo-split graphs. Then, for all $n \in \mathbb{N}$:*

$$f_{\mathcal{G}}(n) = \begin{cases} \lfloor n/2 \rfloor, & \text{if } n \in \{1,2,3,4\}; \\ n, & \text{if } n \in \{5,6\}; \\ n - \min\{k \in \mathbb{N} \mid 2^k + k > n - 6\}, & \text{if } n \geq 7. \end{cases}$$

Consequently, for all $n \in \mathbb{N}$ we have $n - \lfloor \log n \rfloor - 1 \leq f_{\mathcal{G}}(n) \leq n - \lfloor \log n \rfloor + 2$.

Proof. Let us denote by $\mathcal{S}$ the class of split graphs. For all $n \leq 4$, every n-vertex pseudo-split graph is a split graph and consequently $f_{\mathcal{G}}(n) = f_{\mathcal{S}}(n)$. Recall from Theorem 5.3.2 that $f_{\mathcal{S}}(n) = n - \min\{k \in \mathbb{N} \mid 2^k + k > n\}$. Hence, $f_{\mathcal{S}}(1) = 0$, $f_{\mathcal{S}}(2) = f_{\mathcal{S}}(3) = 1$, $f_{\mathcal{S}}(4) = 2$. It follows that $f_{\mathcal{S}}(n) = \lfloor n/2 \rfloor$.

Let now $n \in \{5, 6\}$ and let G be a pseudo-split graph on n vertices. If $n = 5$, then either $S = \emptyset$ and G is a split graph, or $G \cong C_5$. If G is a split graph, then $s(G) \leq f_{\mathcal{S}}(5) = 3$, while for $G \cong C_5$ we have $s(G) = 5$. It follows that $f_{\mathcal{G}}(5) = 5$. Similarly, if $n = 6$, then either $S = \emptyset$ and G is a split graph with $s(G) \leq f_{\mathcal{S}}(6) = 3$, or $G \cong C_5 * P_1$ with $s(G) = 5$, or $G \cong C_5 + P_1$, with $s(G) = 6$. We conclude that $f_{\mathcal{G}}(6) = 6$.

For all $n \geq 7$, let $g(n)$ be the function defined as follows:

$$g(n) = \max\{f_{\mathcal{S}}(n), f_{\mathcal{S}}(n-5) + 5, f_{\mathcal{S}}(n-6) + 6\}.$$

Let $n \geq 7$ be arbitrary. We want to show that $f_{\mathcal{G}}(n) = n - \min\{k \in \mathbb{N} \mid 2^k + k > n - 6\}$. First we will prove that $g(n) = f_{\mathcal{S}}(n-6) + 6$. Then we will prove that $f_{\mathcal{G}}(n) = g(n)$. By Theorem 5.3.2, this will imply the explicit formula given by Theorem 5.3.3.

In order to prove that

$$g(n) = \max\{f_{\mathcal{S}}(n), f_{\mathcal{S}}(n-5) + 5, f_{\mathcal{S}}(n-6) + 6\} = f_{\mathcal{S}}(n-6) + 6,$$

it suffices to show that $f_{\mathcal{S}}(n) \leq f_{\mathcal{S}}(n-i) + i$ for all $i \in \{0, \ldots, n-1\}$, as that will imply that $f_{\mathcal{S}}(n) \leq f_{\mathcal{S}}(n-5) + 5 \leq f_{\mathcal{S}}(n-6) + 6$. Let $i \in \{0, \ldots, n-1\}$ and let

$m = n-i$. The inequality $f_{\mathcal{S}}(n) \leq f_{\mathcal{S}}(n-i)+i$ is equivalent to $f_{\mathcal{S}}(m+i) \leq f_{\mathcal{S}}(m)+i$. By Theorem 5.3.2 this is equivalent to:

$$m + i - \min\{k \in \mathbb{N} \mid 2^k + k > m + i\} \leq m - \min\{k \in \mathbb{N} \mid 2^k + k > m\} + i,$$

which simplifies to $\min\{k \in \mathbb{N} \mid 2^k + k > m\} \leq \min\{k \in \mathbb{N} \mid 2^k + k > m + i\}$. The last inequality holds since the function $\kappa(m) = \min\{k \in \mathbb{N} \mid 2^k + k > m\}$ is clearly non-decreasing. Therefore, $g(n) = f_{\mathcal{S}}(n-6)+6$, as claimed.

Now we prove that $f_{\mathcal{G}}(n) = g(n)$. We divide the proof into two parts, by proving each of the two inequalities. First we prove that $f_{\mathcal{G}}(n) \geq g(n)$. Let G' be a split graph with $n-6$ vertices such that $s(G') = f_{\mathcal{S}}(n-6)$ and let (C, I) be a split partition of G' such that C is a maximal clique in G'. Let G'' be the disjoint union of G', C_5, and P_1. Let us denote the vertex set of the C_5 by S and the unique vertex of P_1 by v. We add to G'' all the edges joining a vertex in C with a vertex in $S \cup \{v\}$. The resulting graph G has n vertices and is a pseudo-split graph, with a pseudo-split partition $(C, I \cup \{v\}, S)$. By Lemma 5.3.1, every minimal separator in G' is the neighborhood of some vertex in I. It is not difficult to see that G' is an induced subgraph of G and that every minimal separator in G' is also a minimal separator in G. Moreover, the neighborhood of vertex v separates v and any vertex in S, so we have that $C = N_G(v) \in \mathcal{S}_G$, while $C \notin \mathcal{S}_{G'}$, by Lemma 5.3.1. Similarly, we infer that for all $u \in S$ it holds that $N_G(u) \in \mathcal{S}_G \backslash \mathcal{S}_{G'}$. Since all vertices in S have distinct neighborhoods in G, they define five distinct separators in G. It follows that $s(G) \geq s(G')+6$. Since $f_{\mathcal{G}}(n) \geq s(G) \geq s(G')+6 = f_{\mathcal{S}}(n-6)+6$, we infer that $f_{\mathcal{G}}(n) \geq g(n)$, as claimed.

Next, we show that $f_{\mathcal{G}}(n) \leq g(n)$. Let G be an n-vertex pseudo-split graph, with a pseudo-split partition (C, I, S). We will show that $s(G) \leq g(n)$. Since this will be true for an arbitrary n-vertex pseudo-split graph, this will imply that $f_{\mathcal{G}}(n) \leq g(n)$. If $S = \emptyset$, then G is a split graph and $s(G) \leq f_{\mathcal{S}}(n) \leq g(n)$. So let $S \neq \emptyset$. Let $U = \{v \in I \mid N(v) = C\}$, $I' = I \setminus U$, and $k = |U|$. Then the subgraph G' of G induced by $C \cup I'$ is a split graph with $n-k-5$ vertices and thus $s(G') \leq f_{\mathcal{S}}(n-k-5)$. Moreover, by the definition of U, we infer that C is a maximal clique in G'. By Lemma 5.3.1 it follows that $\mathcal{S}_{G'} = \{N_{G'}(u) \mid u \in I'\}$. We now describe all minimal separators in G. Since G is $2P_2$-free, by Lemma 3.1.3 every minimal separator in G is the neighborhood of some vertex. Let $u \in V(G)$. If $u \in I$, then $N_G(u)$ is a minimal separator in G, since it separates u and any vertex v in S. If $u \in S$, then $N_G(u)$ separates u from any vertex in S that is non-adjacent to u. If $u \in C$, then the subgraph of G induced by $\{u\}$ is the only $N_G(u)$-full component in $G - N_G(u)$, so $N_G(u)$ cannot be a minimal separator in G. It follows that $\mathcal{S}_G = \{N_G(u) \mid u \in I \cup S\}$. Consequently, if $k = 0$, then $\mathcal{S}_G = \mathcal{S}_{G'} \cup \{N_G(u) \mid u \in S\}$ and

$$s(G) = s(G') + 5 \leq f_{\mathcal{S}}(n-5) + 5 \leq g(n),$$

while if $k \geq 1$, then $\mathcal{S}_G = \mathcal{S}_{G'} \cup \{N_G(u) \mid u \in S\} \cup \{C\}$ and

$$s(G) = s(G') + 6 \leq f_{\mathcal{S}}(n-k-5) + 6 \leq f_{\mathcal{S}}(n-6) + 6 = g(n).$$

It remains to show that for all $n \in \mathbb{N}$, we have $n - \lfloor \log n \rfloor - 1 \leq f_{\mathcal{G}}(n) \leq n - \lfloor \log n \rfloor + 2$. For $n \in \{1, \ldots, 6\}$, this can be easily verified. For $n \geq 7$, the inequalities are equivalent to proving that $\lfloor \log n \rfloor - 2 \leq k \leq \lfloor \log n \rfloor + 1$, where k is the smallest positive integer such that $2^k + k > n-6$. The definition of k implies that $2^{k-1} + k - 1 \leq n-6$. Consequently $k-1 \leq \log(2^{k-1}+k-1) \leq \log(n-6) \leq \log n$, which implies $k \leq \log n + 1$

and thus, since k is an integer, $k \leq \lfloor \log n \rfloor + 1$. To prove that $k \geq \lfloor \log n \rfloor - 2$, note that $n \leq 2^k + k + 5 \leq 4 \cdot 2^k = 2^{k+2}$, which implies $\lfloor \log n \rfloor \leq \log n \leq k + 2$. This completes the proof. $\square$

Finally, we consider the class of $2P_2$-free graphs. The following lemma (originally stated in Chapter 3) gives a necessary condition for a minimal separator in a $2P_2$-free graph (see [184]).

Lemma 3.1.3. *Let G be a $2P_2$-free graph and let S be a minimal separator in G. Then there exists a vertex $v \in V(G)$ such that $S = N(v)$.*

From the above lemma it follows that for any $n \in \mathbb{N}$ we have that $f_{\mathcal{G}}(n) \leq n$, where $\mathcal{G}$ is a class of $2P_2$-free graphs. The following result will be useful to establish the lower bound.

Theorem 5.3.4. *Let G be a $2P_2$-free graph and let v be a vertex in G such that for every $w \in V(G)$, if $N_G(w) \subseteq N_G(v)$ or $N_G[w] \subseteq N_G[v]$, then w is isolated in G. Then $N_G(v) \in \mathcal{S}_G$.*

Proof. Let G be a $2P_2$-free graph and let v be a vertex in G. Assume that for every vertex w in G, $w \neq v$, it holds that $N_G(w) \nsubseteq N_G(v)$ and $N_G[w] \nsubseteq N_G[v]$. We want to prove that $N_G(v) \in \mathcal{S}_G$.

Let $C_1, \ldots, C_k$ be connected components of $G - N_G[v]$. Assume for contradiction that $N_G(v)$ is not a minimal separator in G. Then it holds that for every component C_i, $i \in \{1, \ldots, k\}$, there exists some vertex $u \in N_G(v)$ such that $N_G(u) \cap C_i = \emptyset$.

Since the closed neighborhoods of vertex v and all other vertices in G are incomparable, it follows that every vertex in $N_G(v)$ has a neighbor in the set $V(G) \setminus N[v]$. Let C be the component from $\{C_1, \ldots, C_k\}$ that maximizes the number $|\{N_G(u) \cap C_i \mid u \in N_G(v)\}|$ among all $i \in \{1, \ldots, k\}$. Observe that $|\{N_G(u) \cap C \mid u \in N_G(v)\}| \geq 1$. As $N_G(v)$ is not minimal separator in G, we have that C cannot be $N_G(v)$-full component, so there exists a vertex $u \in N_G(v)$ having no neighbor in C. Since $N_G[u] \nsubseteq N_G[v]$, it follows that there exists some vertex $w \in C_j$, $j \in \{1, \ldots, k\}$, $C \neq C_j$, such that $uw \in E(G)$. If the component C has at least two vertices, then there exists some edge e in C and together with edge uw it obtains a forbidden $2P_2$. So it follows that C contains just one vertex, say v'. But then $N_G(v') \subseteq N_G(v)$, contrary to the assumption. This yields a contradiction, so it cannot be true that $N_G(u) \cap C = \emptyset$, and thus $N_G(N_G(v)) = C$, implying that S is a minimal separator in G, as we wanted to prove. $\square$

Corollary 5.3.5. *Let $\mathcal{G}$ be a class of graphs containing complements of cycles of length at least 5. Then $f_{\mathcal{G}}(n) \geq n$, for all $n \geq 5$.*

Proof. Let $n \geq 5$. Then the graph $G_n = \overline{C_n}$ is in $\mathcal{G}$ and we have that $f_{\mathcal{G}}(n) \geq s(G_n)$. Let $u, v \in V(G_n)$ be distinct vertices in G_n. Then neither their neighborhoods, nor closed neighborhoods are comparable, so by Theorem 5.3.4, it follows that for every vertex $v \in V(G_n)$, the set $N_{G_n}(v)$ is a minimal separator in G_n. Since $|V(G_n)| = n$, it follows that $s(G_n) = n$ and thus $f_{\mathcal{G}}(n) \geq n$. $\square$

Theorem 5.3.6. *Let $\mathcal{G}$ be the class of $2P_2$-free graphs. Then, for all $n \in \mathbb{N}$:*

$$f_{\mathcal{G}}(n) = \begin{cases} n-1, & \text{if } n \in \{1,2\}; \\ n-2, & \text{if } n \in \{3,4\}; \\ n, & \text{if } n \geq 5. \end{cases}$$

Proof. Lemma 3.1.3 implies that $f_{\mathcal{G}}(n) \leq n$ for all $n \in \mathbb{N}$. We first show that for every two non-negative integers k, ℓ, not both equal 0, we have $f_{\mathcal{G}}(n) \geq n$ for $n = 5k + 6\ell$. Since every $n \geq 20$ can be written in such a way [223], this will imply that $f_{\mathcal{G}}(n) = n$ for all $n \geq 20$. Let thus $n = 5k + 6\ell$ with $k, \ell \in \mathbb{Z}_+$ and let G be the join of k copies of C_5 and ℓ copies of the graph $C_5 + P_1$. Since the complement of G is the disjoint union of k copies of C_5 and ℓ copies of the join $C_5 * P_1$, the complement of G is C_4-free. It follows that G is an n-vertex $2P_2$-free graph. Since $s(C_5) = 5$, by Corollary 3.2.2 it follows that $s(C_5 + P_1) = 6$. Hence, applying Corollary 3.2.3, we get $f_G(n) \geq s(G) = ks(C_5) + \ell s(C_5 + P_1) = 5k + 6\ell = n$, as claimed. Let us now analyze the cases when $n \in \{1, \ldots, 19\}$. A straightforward verification shows that $f_{\mathcal{G}}(n) = n - 1$ for $n \in \{1, 2\}$ and $f_{\mathcal{G}}(n) = n - 2$ for $n \in \{3, 4\}$. Let now $n \geq 5$ and let n' be the largest integer such that $n' \leq n$ and $n' = 5k + 6\ell$ for some $k, \ell \in \mathbb{Z}_+$. If $n \in \{5, 6, 10, 11, 12, 15, 16, 17, 18\}$, then it is not difficult to see that $n' = n$, hence we can use the above construction to show that $f_{\mathcal{G}}(n) = n$. If $n \in \{7, 8, 9\}$, then the graph $(C_5 + P_1) * ((n - 6)P_1)$ shows that $f_{\mathcal{G}}(7) \geq 6$, $f_{\mathcal{G}}(8) \geq 7$, and $f_{\mathcal{G}}(9) \geq 7$. If $n = 13$ (resp., $n = 19$), then the graph G obtained from the disjoint union of P_1 and the join of $\ell = 2$ (resp., $\ell = 3$) copies of $C_5 + P_1$ is an n-vertex $2P_2$-free graph with $s(G) = \ell s(C_5 + P_1) + 1 = 6\ell + 1 = n$ (by Corollary 3.2.2). For $n = 14$, adding an isolated vertex to the above 13-vertex graph shows that $f_{\mathcal{G}}(n) \geq n - 1$ in this case. This completes the proof. $\square$

Theorem 5.3.7. *Let $\mathcal{G}$ be the class of $\{2P_2, C_5\}$-free graphs. Then, for all $n \in \mathbb{N}$ we have:*

$$f_{\mathcal{G}}(n) = \begin{cases} n-1, & \text{if } n \in \{1,2\}; \\ n-2, & \text{if } n \in \{3,4\}; \\ n, & \text{if } n \geq 5. \end{cases}$$

Proof. If $n \leq 4$, then the classes of $2P_2$-free graphs and $\{2P_2, C_5\}$-free graphs coincide. Let thus $n \geq 5$. Every graph in $\mathcal{G}$ is $2P_2$-free, so we have that $f_{\mathcal{G}}(n) \leq n$. Moreover, the graph on n vertices that is isomorphic to the complement of the n-cycle belongs to the class $\mathcal{G}$, so we have that $f_{\mathcal{G}}(n) \geq s(\overline{C_n})$ and by Corollary 5.3.5, we have that $f_{\mathcal{G}}(n) \geq s(\overline{C_n}) \geq n$. Finally, it follows that $f_{\mathcal{G}}(n) = n$, as we wanted to show. $\square$

Chapter 6

Bisimplicial separators

Recall that the class of chordal graphs is exactly the class of graphs where all minimal separators are cliques [95]. In this chapter we study a generalization of this concept. For a class $\mathcal{C}$ of graphs, we denote by $\mathcal{G}_{\mathcal{C}}$ the class of all graphs G such that every minimal separator of G induces a graph from $\mathcal{C}$. For an integer $k \geq 0$, we denote by $\mathcal{G}_k$ the class of all graphs G that have the property that every minimal separator of G is a union of at most k cliques. Note that $\mathcal{G}_0$ is the class of all disjoint unions of (arbitrarily many) complete graphs, and (by [95]) $\mathcal{G}_1$ is the class of all chordal graphs. We give structural and algorithmic results for classes $\mathcal{G}_k$, summarized as follows:

(1) We show that for any hereditary graph class $\mathcal{C}$ that is closed under edge addition, both the class $\mathcal{C}$ and the corresponding class $\mathcal{G}_{\mathcal{C}}$ are closed under induced minors, and we characterize the class $\mathcal{G}_{\mathcal{C}}$ in terms of forbidden induced minors. In particular, we give a complete list of minimal forbidden induced minors for the class $\mathcal{G}_2$.

(2) We generalize results on chordal graphs(by Dirac [95] and by Rose, Tarjan, and Lueker [212]) by showing that for every positive integer k, every LexBFS ordering of a graph in $\mathcal{G}_k$ is a k-simplicial elimination ordering. This implies that every nonnull graph in $\mathcal{G}_k$ has a k-simplicial vertex (i.e., a vertex whose neighborhood is a union of k cliques).

(3) We show that for each $k \geq 3$, it is NP-hard to recognize graphs in $\mathcal{G}_k$. (Graph in $\mathcal{G}_1$ can be recognized in polynomial time, while the time complexity of recognizing graphs in $\mathcal{G}_2$ is unknown.)

(4) We show the decomposition theorems and polynomial-time recognition for various subclasses of $\mathcal{G}_2$.

(5) We construct various polynomial-time algorithms in class $\mathcal{G}_2$ or in its subclasses (see Table 6.1 on page 94).

Results presented in this chapter are based on the following paper: [180] M. Milanič, I. Penev, N. Pivač, K. Vušković, Bisimplicial separators. J.Graph Theory, 2024. doi:10.1002/jgt.23098.

6.1 Introduction and preliminary results

One can *define* a graph class by restricting the properties of minimal separators of graphs in the class. Given a class $\mathcal{C}$ of graphs, we denote by $\mathcal{G}_{\mathcal{C}}$ the class of all graphs G such that every minimal separator of G induces a graph from $\mathcal{C}$. Since complete graphs have no separators, we see for all classes $\mathcal{C}$, the class $\mathcal{G}_{\mathcal{C}}$ contains all complete graphs (including the null graph). In this work we restrict our attention to classes $\mathcal{C}_k$ consisting of graphs whose vertex sets can be partitioned into k (possibly empty) cliques. This naturally leads to the definition of a graph class $\mathcal{G}_{\mathcal{C}_k}$, that consists of graphs whose minimal separators can be partitioned into k (possibly empty) cliques. For an integer $k \geq 0$, we say that a minimal separator is *k-simplicial* if it is a union of at most k cliques and we denote by $\mathcal{G}_k = \mathcal{G}_{\mathcal{C}_k}$ the class of all graphs in which every minimal separator is k-simplicial. If $k = 1$ (resp., $k = 2$), we simplify out terminology, so that k-simplicial becomes *simplicial* (resp., *bisimplicial*).

Note that $\mathcal{G}_0$ is the class of all disjoint unions of (arbitrarily many) complete graphs, and (by [95]) $\mathcal{G}_1$ is the class of all chordal graphs. Furthermore, for every $k \geq 0$ it holds that $\mathcal{G}_k \subseteq \mathcal{G}_{k+1}$ and all these inclusions are proper, as verified by the class of complete bipartite graphs with exactly two vertices in one of the two parts. We provide some characterizations of the class $\mathcal{G}_{\mathcal{C}}$ when $\mathcal{C}$ is hereditary, that is, closed under vertex deletion. Note that if $\mathcal{G}_{\mathcal{C}}$ is hereditary, then the class $\mathcal{C}$ is contained in the class $\mathcal{G}_{\mathcal{C}}$.

Proposition 6.1.1. *Let $\mathcal{C}$ be a hereditary class of graphs, and let G be a graph. Then the following are equivalent:*

(1) $G \in \mathcal{G}_{\mathcal{C}}$;

(2) for all induced subgraphs H of G, every minimal separator S of H satisfies $H[S] \in \mathcal{C}$;

(3) for all induced subgraphs H of G, every minimal cutset C of H satisfies $H[C] \in \mathcal{C}$.

Proof. We prove the result by showing that (1) implies (2), that (2) implies (3), and that (3) implies (1).

First, we assume that (1) holds, and we prove (2). Let H be an induced subgraph of G, and suppose that $a, b \in V(H)$ are distinct, non-adjacent vertices, and that S is a minimal (a, b)-separator of H. Then $S \cup (V(G) \setminus V(H))$ is an (a, b)-separator of G. Let $S^* \subseteq S \cup (V(G) \setminus V(H))$ be a minimal (a, b)-separator of G; by (a), $G[S^*] \in \mathcal{C}$. Since S^* is an (a, b)-separator of G, we have that $S^* \cap V(H)$ is an (a, b)-separator of H. Moreover, $S^* \cap V(H) \subseteq S$, and so the minimality of S guarantees that $S^* \cap V(H) = S$; consequently, $S \subseteq S^*$. Since $G[S^*] \in \mathcal{C}$, and since $\mathcal{C}$ is hereditary, it follows that $G[S] \in \mathcal{C}$. Clearly, $G[S] = H[S]$, and so $H[S] \in \mathcal{C}$. Thus, (2) holds.

Next, we assume that (2) holds, and we prove (3). Let H be an induced subgraph of G, and suppose that C is a minimal cutset of H. Let A and B be the vertex sets of two distinct components of $H \setminus C$. The minimality of C guarantees that every vertex in C has a neighbor both in A and in B, and this, in turn, guarantees that for all $a \in A$ and $b \in B$, C is a minimal (a, b)-separator of H. But now (2) implies that $H[C] \in \mathcal{C}$. Thus, (3) holds.

Finally, we assume that (3) holds, and we prove (1). Suppose that $a, b \in V(G)$ are distinct, non-adjacent vertices, and that S is a minimal (a, b)-separator of G. Let A (resp. B) be the vertex set of the component of $G \setminus S$ that contains a (resp. b). Clearly,

A and B are disjoint and anticomplete to each other. Furthermore, the minimality of S implies that every vertex in S has a neighbor both in A and in B. Set $H := G[A \cup B \cup S]$. Then (A, B, S) is a cut-partition of H; furthermore, since $H[A]$ and $H[B]$ are connected, and every vertex of S has a neighbor both in A and in B, we see that S is a minimal cutset of H. Now (3) guarantees that $H[S] \in \mathcal{C}$; since $H[S] = G[S]$, it follows that $G[S] \in \mathcal{C}$. Thus, (1) holds. □

The equivalence of conditions (1) and (2) in the above theorem implies the following.

Corollary 6.1.2. *Let $\mathcal{C}$ be a hereditary class of graphs. Then $\mathcal{G}_{\mathcal{C}}$ is hereditary.*

Proof. This readily follows from Proposition 6.1.1, and more precisely, from the equivalence of (1) and (2) from Proposition 6.1.1. □

Recall that the operation of gluing H_1 and H_2 along a clique produces a graph obtained from $H_1 \cup H_2$ by choosing cliques C_1 in H_1 and C_2 in H_2 such that $|C_1| = |C_2|$, fixing a bijection f from C_1 to C_2, and identifying each vertex $v \in C_1$ with the vertex $f(v)$.

Theorem 6.1.3. *Let $\mathcal{C}$ be a hereditary class of graphs that contains all complete graphs.*[1] *Then, $\mathcal{G}_{\mathcal{C}}$ is closed under gluing along a clique.*

Proof. Let G be a graph that admits a clique cutset C and let (A, B, C) be an associated cut-partition of G. Assume that $G_A := G[A \cup C]$ and $G_B := G[B \cup C]$ both belong to $\mathcal{G}_{\mathcal{C}}$. It suffices to show that $G \in \mathcal{G}_{\mathcal{C}}$.

Fix a pair of distinct, non-adjacent vertices $x, y \in V(G)$, and let S be a minimal (x, y)-separator of G. We must show that $G[S] \in \mathcal{C}$. Since C is a clique, it contains at most one of x, y; by symmetry, we may therefore assume that $x \in A$. We now consider two cases: when $y \in A \cup C$, and when $y \in B$.

Case 1: $y \in A \cup C$. Then $S \cap (A \cup C)$ is an (x, y)-separator of G_A; let $S' \subseteq S \cap (A \cup C)$ be a minimal (x, y)-separator of G_A. Since $G_A \in \mathcal{G}_{\mathcal{C}}$, we see that $G_A[S'] = G[S']$ belongs to $\mathcal{C}$. If $S' = S$, then we are done. So, assume that $S' \subsetneqq S$. The minimality of S then implies that there is a path $p_1, \dots, p_s$ in $G \setminus S'$, with $p_1 = x$ and $p_s = y$. Since S' is an (x, y)-separator of G_A, we see that at least one vertex of the path $p_1, \dots, p_s$ belongs to B. Let i be the smallest index in $\{1, \dots, s\}$ such that $p_i \in B$, and let j be the largest index in $\{1, \dots, s\}$ such that $p_j \in B$. Since $p_1, p_s \in A \cup C$, we have that $2 \le i \le j \le s-1$. Moreover, since A is anticomplete to B, we have that $p_{i-1}, p_{j+1} \in C$; since C is a clique, we see that p_{i-1}, p_{j+1} are adjacent. But now $p_1, \dots, p_{i-1}, p_{j+1}, \dots, p_s$ is a path between x and y in $G_A \setminus S'$, contrary to the fact that S' is an (x, y)-separator of G_A.

Case 2: $y \in B$. Note that, in this case, C is an (x, y)-separator of G.

Claim. At least one of $S \cap A$ and $S \cap B$ is empty.

Proof of the Claim. Suppose otherwise, i.e., that both $S \cap A$ and $S \cap B$ are nonempty. Set $S_A := S \setminus B$ and $S_B := S \setminus A$. By the minimality of S, there is a path $p_1, \dots, p_s$, with $p_1 = x$ and $p_s = y$, in $G \setminus S_A$, and there is a path $q_1, \dots, q_t$, with $q_1 = x$ and $q_t = y$, in $G \setminus S_B$. Since $p_1 \in A$ and $p_s \in B$, and since A is anticomplete to B, some internal vertex of the path $p_1, \dots, p_s$ belongs to C; let i be the smallest index in $\{2, \dots, s-1\}$

[1] However, not all graphs in $\mathcal{C}$ have to be complete.

such that $p_i \in C$ (then $p_1, \dots, p_{i-1} \in A$). Similarly, at least one internal vertex of the path $q_1, \dots, q_t$ belongs to C; let j be the largest index in $\{2, \dots, t-1\}$ such that $q_j \in C$ (then $q_{j+1}, \dots, q_t \in B$). Since C is a clique, we see that p_i and q_j are either equal or adjacent. In the former case, $p_1, \dots, p_i, q_{j+1}, \dots, q_t$ is a path between x and y in $G \setminus S$; and in the latter case, $p_1, \dots, p_i, q_j, \dots, q_t$ is a path between x and y in $G \setminus S$. But neither outcome is possible, since S is an (x, y)-separator of G. This proves the Claim. ♦

By the Claim, and by symmetry, we may assume that $S \cap B = \emptyset$, i.e., $S \subseteq A \cup C$. Let Y be the vertex set of the component of $G[B]$ that contains y.

Suppose first that $C \setminus S$ is anticomplete to Y. Then $C \cap S$ is an (x, y)-separator of G, and so the minimality of S guarantees that $S \subseteq C$. Thus, S is a clique; since $\mathcal{C}$ contains all complete graphs, it follows that $G[S] \in \mathcal{C}$, and we are done.

From now on, we assume that $C \setminus S$ is not anticomplete to Y. Fix a vertex $c \in C \setminus S$ that has a neighbor in Y. Since $y \in Y$, and $G[Y]$ is connected, the graph G contains a path $q_1, \dots, q_t$, with $q_1 = c$, $q_t = y$, and $q_2, \dots, q_t \in Y$ (so, $q_2, \dots, q_t \in B$). Now, suppose that there is a path $p_1, \dots, p_s$ in $G_A \setminus S$, with $p_1 = x$ and $p_s = c$. Then $p_1, \dots, p_s, q_2, \dots, q_t$ is a path in $G \setminus S$ between x and y, contrary to the fact that S is an (x, y)-separator of G. So, S is an (x, c)-separator of G_A. Let $S' \subseteq S$ be a minimal (x, c)-separator of G_A; since $G_A \in \mathcal{G}_{\mathcal{C}}$, we see that $G_A[S'] = G[S']$ belongs to $\mathcal{C}$. If S' is an (x, y)-separator of G, then the minimality of S guarantees that $S = S'$, and we are done. So, assume that S' is not an (x, y)-separator of G. Then there is a path $r_1, \dots, r_k$ in $G \setminus S'$, with $r_1 = x$ and $r_k = y$. Since $x \in A$, $y \in B$, and A is anticomplete to B, we see that some internal vertex of $r_1, \dots, r_k$ belongs to C; let i be the smallest index in $\{2, \dots, k-1\}$ such that $r_i \in C$. Since $r_i, c \in C$, and C is a clique, we see that r_i and c are either equal or adjacent. In the former case, $r_1, \dots, r_i$ is a path from x to c in $G_A \setminus S'$, and in the latter case, $r_1, \dots, r_i, c$ is a path from x to c in $G_A \setminus S'$. But neither outcome is possible, since S' is an (x, c)-separator of G_A. □

Theorem 6.1.3 implies the following result on the classes of graphs in which every minimal separator is a union of k cliques.

Corollary 6.1.4. *For every positive integer k, the class $\mathcal{G}_k$ is closed under gluing along a clique.*

Proof. This follows immediately from Theorem 6.1.3, the fact that $\mathcal{G}_k = \mathcal{G}_{\mathcal{C}_k}$, and the fact that the class $\mathcal{C}_k$ is a hereditary graph class containing all complete graphs. □

Note that Corollary 6.1.4 fails for $\mathcal{G}_0$: the two-edge path P_3 is an obvious counterexample.

6.2 Forbidden induced minors

For a class of graphs $\mathcal{C}$, let us denote by $\mathcal{M}_{\mathcal{C}}$ the class of all graphs that do not belong to $\mathcal{C}$, but all of whose proper induced minors do belong to $\mathcal{C}$. Note that nonisomorphic graphs in $\mathcal{M}_{\mathcal{C}}$ are incomparable under the induced minor relation. If the class $\mathcal{C}$ is closed under induced minors, then clearly, $\mathcal{C}$ is precisely the class of all $\mathcal{M}_{\mathcal{C}}$-induced-minor-free graphs. In this case, we refer to graphs in $\mathcal{M}_{\mathcal{C}}$ as the *minimal forbidden induced minors* for the class $\mathcal{C}$. More generally, if $\mathcal{M}$ is a class of graphs such that every graph in $\mathcal{C}$ is $\mathcal{M}$-induced-minor-free, we refer to graphs in $\mathcal{M}$ as *forbidden induced minors* for the class $\mathcal{C}$.

We now consider hereditary graph classes that are closed under edge addition. We show that for any such class $\mathcal{C}$, both the class $\mathcal{C}$ and the corresponding class $\mathcal{G}_{\mathcal{C}}$ are closed under induced minors, and we characterize the class $\mathcal{G}_{\mathcal{C}}$ in terms of forbidden induced minors. Recall that $H_1 * H_2$ denotes the join of graphs H_1 and H_2 on disjoint vertex sets.

Theorem 6.2.1. *Let $\mathcal{C}$ be a hereditary class of graphs, closed under edge addition. Then both $\mathcal{C}$ and $\mathcal{G}_{\mathcal{C}}$ are closed under induced minors, and $\mathcal{G}_{\mathcal{C}}$ is precisely the class of all $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$-induced-minor-free graphs.*

Proof. Let $\mathcal{C}$ be a hereditary class of graphs, closed under edge addition. First, we prove that $\mathcal{C}$ is closed under induced minors. It suffices to show that $\mathcal{C}$ is closed under vertex deletion and edge contraction. The former follows immediately from the fact that $\mathcal{C}$ is hereditary. Let us prove the latter. Fix a graph $G \in \mathcal{C}$, and fix an edge $xy \in E(G)$. Then G/xy is isomorphic to the graph obtained from G by first deleting y, and then adding edges between x and all vertices in $N_G[y] \setminus N_G[x]$; since $\mathcal{C}$ is closed under vertex deletion and edge addition, it follows that G/xy belongs to $\mathcal{C}$. So, $\mathcal{C}$ is closed under edge contraction. This proves that $\mathcal{C}$ is closed under induced minors.

By Corollary 6.1.2, $\mathcal{G}_{\mathcal{C}}$ is hereditary. So, in order to prove that $\mathcal{G}_{\mathcal{C}}$ is closed under induced minors, it suffices to show that $\mathcal{G}_{\mathcal{C}}$ is closed under edge contractions. Fix $G \in \mathcal{G}_{\mathcal{C}}$, let xy be an edge of G, and set $G' := G/xy$; the vertex of G' to which the edge xy is contracted will be denoted by v^{xy}. We must show that $G' \in \mathcal{G}_{\mathcal{C}}$, i.e., that for any minimal separator S of G', we have that $G'[S] \in \mathcal{C}$.

We first deal with minimal separators of G' that contain v^{xy}. So, suppose that $S \subseteq V(G')$ is a minimal separator of G' such that $v^{xy} \in S$; we must show that $G'[S] \in \mathcal{C}$. Fix distinct $a, b \in V(G') \setminus S$ such that S is a minimal (a, b)-separator of G'. Then $S^* := (S \setminus \{v^{xy}\}) \cup \{x, y\}$ is an (a, b)-separator of G. Let $S' \subseteq S^*$ be a minimal (a, b)-separator of G; since $G \in \mathcal{G}_{\mathcal{C}}$, we have that $G[S'] \in \mathcal{C}$. If $x, y \notin S'$, then $S' \subsetneqq S$ is an (a, b)-separator of G', contrary to the minimality of S. So, S' contains at least one of x, y. Then $(S' \setminus \{x, y\}) \cup \{v^{xy}\}$ is an (a, b)-separator of G'; since $(S' \setminus \{x, y\}) \cup \{v^{xy}\} \subseteq S$, the minimality of S implies that $S = (S' \setminus \{x, y\}) \cup \{v^{xy}\}$. As we show next, the graph $G'[S]$ can be obtained from an induced subgraph of $G[S']$ by possibly adding some edges. By symmetry, we may assume that $x \in S'$. Since $S \setminus \{v^{xy}\} = S' \setminus \{x, y\}$, the graph $G'[S]$ is isomorphic to the graph obtained from the subgraph of $G[S']$ induced by $S' \setminus \{y\}$ by adding to it the edges from x to all vertices in $S' \setminus \{x, y\}$ that are adjacent in G to y but not to x. Since $G[S'] \in \mathcal{C}$, and $\mathcal{C}$ is hereditary and closed under edge addition, we deduce that $G'[S] \in \mathcal{C}$, and we are done.

We still have to consider minimal separators of G' that do not contain v^{xy}. Here, we first observe that for any pair of non-adjacent vertices a, b of G', and any set $S \subseteq V(G') \setminus \{a, b, v^{xy}\} = V(G) \setminus \{a, b, x, y\}$, both the following hold:

(1) if $v^{xy} \notin \{a, b\}$, then S is an (a, b)-separator of G if and only if S is an (a, b)-separator of G';

(2) if $v^{xy} = a$, then S is an (x, b)-separator of G if and only if S is an (a, b)-separator of G'.

Clearly, (1) and (2) imply that any set $S \subseteq V(G') \setminus \{v^{xy}\} = V(G) \setminus \{x, y\}$ is a minimal separator of G' if and only if it is a minimal separator of G. But for any

$S \subseteq V(G') \setminus \{v^{xy}\} = V(G) \setminus \{x, y\}$, we have that $G'[S] = G[S]$, and moreover, if S is a minimal separator of G, then $G[S] \in \mathcal{C}$. This shows that if a set $S \subseteq V(G') \setminus \{v^{xy}\}$ is a minimal separator of G', then $G'[S] \in \mathcal{C}$.

Finally, it remains to show that $\mathcal{G}_{\mathcal{C}}$ is precisely the class of all $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$-induced-minor-free graphs. Let us first show that all graphs in $\mathcal{G}_{\mathcal{C}}$ are $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$-induced-minor-free. Since $\mathcal{G}_{\mathcal{C}}$ is closed under induced minors, it suffices to show that, for all $H \in \mathcal{M}_{\mathcal{C}}$, the graph $2K_1 * H$ does not belong to $\mathcal{G}_{\mathcal{C}}$. So, fix $H \in \mathcal{M}_{\mathcal{C}}$, and let x and y be the two vertices of the $2K_1$ from $2K_1 * H$. Then $V(H)$ is a minimal (x, y)-separator of $2K_1 * H$. Since the subgraph of $2K_1 * H$ induced by $V(H)$ is H, which does not belong to $\mathcal{C}$ (because $H \in \mathcal{M}_{\mathcal{C}}$), it follows that $2K_1 * H$ does not belong to $\mathcal{G}_{\mathcal{C}}$.

For the reverse direction, fix any $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$-induced-minor-free graph G; we must show that $G \in \mathcal{G}_{\mathcal{C}}$. Fix distinct, non-adjacent vertices a and b of G, and fix a minimal (a, b)-separator S of G. We must show that $G[S] \in \mathcal{C}$. Suppose otherwise. Then since $\mathcal{C}$ is closed under induced minors, there exists some $H \in \mathcal{M}_{\mathcal{C}}$ such that H is an induced minor of $G[S]$. We will derive a contradiction by showing that $2K_1 * H$ is an induced minor of G. Let $\{X_v\}_{v \in V(H)}$ be a family of nonempty, pairwise disjoint subsets of S, each inducing a connected subgraph of $G[S]$, and having the property that for all distinct $u, v \in V(H)$, there is an edge between X_u and X_v in $G[S]$ if and only if $uv \in E(H)$. Next, let X_a (resp., X_b) be the vertex set of the component of $G \setminus S$ that contains a (resp., b). Obviously, X_a and X_b are disjoint and anticomplete to each other in G. Further, since S is a minimal (a, b)-separator of G, we see that, in G, every vertex of S has a neighbor both in X_a and in X_b. In particular, for all $v \in V(H)$, there is an edge between X_a and X_v in G, and there is also an edge between X_b and X_v in G. But now by considering the family $\{X_v\}_{v \in \{a,b\} \cup V(H)}$, we see that $2K_1 * H$ is an induced minor of G (here, a and b are the two vertices of the $2K_1$). $\square$

We now apply Theorem 6.2.1 to the cases when $\mathcal{C} = \mathcal{C}_k$ for $k \in \{0, 1\}$, and thus $\mathcal{G}_{\mathcal{C}} = \mathcal{G}_k$.

For $k = 0$, we have $\mathcal{M}_{\mathcal{C}_0} = \{K_1\}$ and therefore $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}_0}\} = \{2K_1 * K_1\} = \{P_3\}$; we obtain that $\mathcal{G}_0$ is precisely the class of all P_3-induced-minor-free graphs. In particular, we recover the known easy fact that the class $\mathcal{G}_0$ of all disjoint unions of complete graphs is precisely the class of all P_3-induced-minor-free graphs (which is also the class of all P_3-free graphs). In this case, since the set of forbidden induced minors is a singleton, it is in fact also the set of *minimal* forbidden induced minors, that is, $\mathcal{M}_{\mathcal{G}_0} = \{P_3\}$.

For $k = 1$, we have $\mathcal{M}_{\mathcal{C}_1} = \{2K_1\}$ and we obtain that $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}_1}\} = \{2K_1 * 2K_1\} = \{C_4\}$; that is, $\mathcal{G}_1$ is precisely the class of all C_4-induced-minor-free graphs. Note that a graph is C_4-induced-minor-free if and only if it contains no induced cycles of length greater than three, that is, it is a chordal graph. By the definition of $\mathcal{G}_1$, graphs in $\mathcal{G}_1$ are precisely the graphs in which all minimal separators are cliques. Thus, we have again recovered a known result: a graph is chordal if and only if all its minimal separators are cliques (see [95]). Since the set of forbidden induced minors is a singleton, we again conclude that $\mathcal{M}_{\mathcal{G}_1} = \{C_4\}$.

The reader may wonder whether, in Theorem 6.2.1, it is necessary to assume that $\mathcal{C}$ is closed under edge addition. Here, we note that if $\mathcal{C}$ is the class of all edgeless graphs, then $K_{2,3} \in \mathcal{G}_{\mathcal{C}}$, but contracting one edge of $K_{2,3}$ produces the diamond, which does not belong to $\mathcal{G}_{\mathcal{C}}$. Thus, the assumption about edge additions cannot be simply removed from Theorem 6.2.1, although it is possible that some other (weaker) assumption would suffice instead.

Corollary 6.2.2. *For all integers $k \geq 0$, classes $\mathcal{C}_k$ and $\mathcal{G}_k$ are closed under induced minors, and all graphs in $\mathcal{G}_k$ are $K_{2,k+1}$-induced-minor-free.*

Proof. Fix an integer $k \geq 0$. Obviously, $\mathcal{C}_k$ is hereditary and closed under edge addition. So, by Theorem 6.2.1, $\mathcal{C}_k$ is closed under induced minors, and $\mathcal{G}_k = \mathcal{G}_{\mathcal{C}_k}$ is closed under induced minors. It remains to show that $K_{2,k+1} \notin \mathcal{G}_k$. But this is obvious: one minimal separator of $K_{2,k+1}$ induces an edgeless subgraph on $k+1$ vertices in $K_{2,k+1}$, and $\mathcal{C}_k$ does not contain edgeless graphs on more than k vertices. □

We now proceed to showing that when $\mathcal{C}$ is a hereditary class of graphs closed not only under edge addition, but also under the addition of universal vertices, the conclusion of Theorem 6.2.1 can be strengthened to a characterization of the class of minimal forbidden induced minors for the class $\mathcal{G}_{\mathcal{C}}$ (see Theorem 6.2.6). Recall that a graph is coconnected if its complement is connected and that every graph is the complete join of its cocomponents.

Proposition 6.2.3. *Let H be a coconnected graph on at least two vertices, and assume that H is an induced minor of a graph G. Let $\{X_v\}_{v \in V(H)}$ be an induced minor model of H in G. Then there exists a cocomponent C of G such that $\bigcup_{v \in V(H)} X_v \subseteq V(C)$.*

Proof.

> **Claim.** For all $v \in V(H)$, there exists a cocomponent C of G such that $X_v \subseteq V(C)$.

Proof of the Claim. Suppose otherwise. Then there exists some vertex $u \in V(H)$ and distinct cocomponents C_1 and C_2 of G such that X_u intersects both $V(C_1)$ and $V(C_2)$. Let us show that u is a universal vertex of H. Fix any $w \in V(H) \setminus \{u\}$. Let C be any cocomponent of G such that $X_w \cap V(C) \neq \emptyset$. By symmetry, we may assume that $C \neq C_1$.[2] Then $X_u \cap V(C_1)$ and $X_w \cap V(C)$ are both nonempty and complete to each other in G, and in particular, there is at least one edge between X_u and X_w in G. So, $uw \in E(H)$. This proves that u is a universal vertex of H. But this is impossible since H is a coconnected graph on at least two vertices, and consequently, H has no universal vertices. ♦

Fix any cocomponent C of G such that $\left(\bigcup_{v \in V(H)} X_v\right) \cap V(C) \neq \emptyset$, and set $U := \{v \in V(H) \mid X_v \cap V(C) \neq \emptyset\}$. By construction, we have that $U \neq \emptyset$, and by the claim, we have that $U = \{v \in V(H) \mid X_v \subseteq V(C)\}$. It now suffices to show that $U = V(H)$, for it will then follow that $\bigcup_{v \in V(H)} X_v \subseteq V(C)$, which is what we need. Since $U \neq \emptyset$ and H is anticonnected, it is in fact enough to show that U is complete to $V(H) \setminus U$ in H. So, fix some $u \in U$ and $w \in V(H) \setminus U$. Since $u \in U$, we have that $X_u \subseteq V(C)$. On the other hand, by the claim, there exists a cocomponent D of G such that $X_w \subseteq V(D)$; since $w \notin U$, we have that $D \neq C$. Since C and D are distinct cocomponents of G, we know that $V(C)$ and $V(D)$ are complete to each other in G; consequently, X_u is complete to X_w in G, and it follows that $uw \in E(H)$. This proves that U is complete to $V(H) \setminus U$, and we are done. □

[2]Indeed, either $C \neq C_1$ or $C \neq C_2$, and by symmetry, we may assume that $C \neq C_1$.

Proposition 6.2.4. *Let H_1 and H_2 be graphs. Assume that H_1 contains no universal vertices and that $2K_1 * H_1$ is an induced minor of $2K_1 * H_2$. Then H_1 is an induced minor of H_2.*

Proof. If H_1 is the null graph, then it is obviously an induced minor of H_2. So, we may assume that H_1 is nonnull. Since $2K_1 * H_1$ is an induced minor of $2K_1 * H_2$, it follows that the graph H_2 is also nonnull.

Now, using the fact that $2K_1 * H_1$ is an induced minor of $2K_1 * H_2$, we fix an induced minor model $\{X_v\}_{v \in V(2K_1 * H_1)}$ of $2K_1 * H_1$ in $2K_1 * H_2$. If $\bigcup_{v \in V(H_1)} X_v \subseteq V(H_2)$, then H_1 is an induced minor of H_2, and we are done. So, we may assume that $\bigcup_{v \in V(H_1)} X_v \not\subseteq V(H_2)$. Fix a cocomponent H of H_1 such that $\bigcup_{v \in V(H)} X_v \not\subseteq V(H_2)$. Since H_1 has no universal vertices, we have that $|V(H)| \geq 2$. In view of Proposition 6.2.3, it follows that $\bigcup_{v \in V(H)} X_v \subseteq V(2K_1)$. Since H has at least two vertices, it follows that $H \cong 2K_1$ and $\bigcup_{v \in V(H)} X_v = V(2K_1)$. Consequently, $\bigcup_{v \in V(2K_1 * H_1) \setminus V(H)} X_v \subseteq V(H_2)$, and see that $(2K_1 * H_1) \setminus V(H)$ is an induced minor of H_2. But note that $(2K_1 * H_1) \setminus V(H) \cong H_1$. So, H_1 is an induced minor of H_2. □

We note that the assumption that H_1 contains no universal vertices cannot be removed from Proposition 6.2.4. To see this, note that $2K_1 * K_2$ is an induced minor of $2K_1 * 3K_1$ (indeed, we obtain $2K_1 * K_2$ by contracting any one edge of $2K_1 * 3K_1$), but K_2 is not an induced minor of $3K_1$.

Lemma 6.2.5. *Let $\mathcal{C}$ be a class of graphs, closed under the addition of universal vertices. Then no graph in $\mathcal{M}_{\mathcal{C}}$ contains a universal vertex. Moreover, for any two nonisomorphic graphs $H_1, H_2 \in \mathcal{M}_{\mathcal{C}}$, the graphs $2K_1 * H_1$ and $2K_1 * H_2$ are incomparable with respect to the induced minor relation.*

Proof. Let us first show that no graph in $\mathcal{M}_{\mathcal{C}}$ contains a universal vertex. Suppose otherwise, and fix a graph $H \in \mathcal{M}_{\mathcal{C}}$ that contains a universal vertex u. By the definition of $\mathcal{M}_{\mathcal{C}}$, we have that $H \setminus u$ belongs to $\mathcal{C}$. But thenc since $\mathcal{C}$ is closed under the addition of universal vertices, we have that $H \in \mathcal{C}$, contrary to the fact that $H \in \mathcal{M}_{\mathcal{C}}$.

Now, fix any two nonisomorphic graphs $H_1, H_2 \in \mathcal{M}_{\mathcal{C}}$. By the definition of $\mathcal{M}_{\mathcal{C}}$, the graphs H_1 and H_2 are incomparable with respect to the induced minor relation. Moreover, by what we just proved, H_1 and H_2 have no universal vertices. So, by Proposition 6.2.4, $2K_1 * H_1$ and $2K_1 * H_2$ are incomparable with respect to the induced minor relation. □

Theorem 6.2.6. *Let $\mathcal{C}$ be a hereditary class of graphs, closed under edge addition, and closed under the addition of universal vertices. Then $\mathcal{M}_{\mathcal{G}_{\mathcal{C}}} = \{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$.*

Proof. By Theorem 6.2.1, $\mathcal{G}_{\mathcal{C}}$ is precisely the class of all $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$-induced-minor-free graphs. On the other hand, Lemma 6.2.5 guarantees that nonisomorphic graphs in $\{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$ are incomparable with respect to the induced minor relation. So, $\mathcal{M}_{\mathcal{G}_{\mathcal{C}}} = \{2K_1 * H \mid H \in \mathcal{M}_{\mathcal{C}}\}$. □

We now apply Theorem 6.2.6 to the cases when $\mathcal{C} = \mathcal{C}_2$, and thus $\mathcal{G}_{\mathcal{C}} = \mathcal{G}_2$. It can be shown that $\mathcal{M}_{\mathcal{G}_2} = \{\overline{K_2 \cup C_{2k+1}} \mid k \in \mathbb{N}\}$ (see Fig. 6.1). To this end, the following auxiliary proposition will be useful.

Proposition 6.2.7. *If a graph H_1 is an induced minor of a graph H_2, then $\overline{H_1}$ is isomorphic to a (not necessarily induced) subgraph of $\overline{H_2}$.*

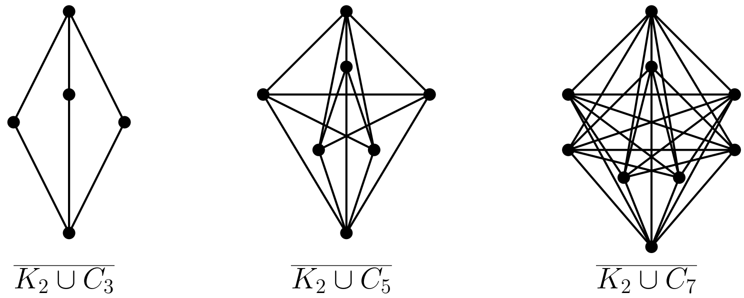


Figure 6.1: Some small graphs in $\mathcal{M}_{\mathcal{G}_2}$.

Proof. The result follows from the following technical claim via simple induction.

> *Claim.* If a graph H is obtained from a graph G by deleting one vertex or contracting one edge, then $\overline{H}$ is isomorphic to a subgraph of $\overline{G}$.

Proof of the Claim. Fix graphs H and G, and assume that H is obtained from G by deleting one vertex or contracting one edge. We must show that $\overline{H}$ is isomorphic to a (not necessarily induced) subgraph of $\overline{G}$. If H is obtained from G by deleting one vertex, then this is obvious. So, assume that H is obtained from G by contracting an edge xy of G. Then $\overline{H}$ is isomorphic to the graph obtained from $\overline{G}$ by first deleting y and then deleting all edges between x and $N_G[y] \setminus N_G[x] = N_{\overline{G}}(x) \setminus N_{\overline{G}}(y)$. So, $\overline{H}$ is isomorphic to a subgraph of $\overline{G}$. ♦ □

Corollary 6.2.8. $\mathcal{M}_{\mathcal{G}_2} = \{\overline{K_2 \cup C_{2k+1}} \mid k \in \mathbb{N}\}$.

Proof. Note that for all positive integers k, we have that $\overline{K_2 \cup C_{2k+1}} \cong 2K_1 * \overline{C_{2k+1}}$. So, in view of Theorem 6.2.6, it is enough to show that $\mathcal{M}_{\mathcal{C}_2} = \{\overline{C_{2k+1}} \mid k \in \mathbb{N}\}$. Note that graphs in $\mathcal{C}_2$ are precisely the complements of bipartite graphs, and it is well known that a graph is bipartite if and only if it contains no odd cycle as an induced subgraph. So, $\mathcal{C}_2$ is precisely the class of all $\{\overline{C_{2k+1}} \mid k \in \mathbb{N}\}$-free graphs. Since $\mathcal{C}_2$ is closed under induced minors (by Corollary 6.2.2), it follows that $\mathcal{C}_2$ is in fact the class of all $\{\overline{C_{2k+1}} \mid k \in \mathbb{N}\}$-induced-minor-free graphs. It remains to show that (nonisomorphic) graphs in $\{\overline{C_{2k+1}} \mid k \in \mathbb{N}\}$ are incomparable with respect to the induced minor relation. But this follows from Proposition 6.2.7, and from the fact that no cycle is a subgraph of a cycle of different length. □

A graph is 1-*perfectly-orientable* if it admits an orientation in which the out-neighborhood of each vertex is a clique of the underlying graph. It was shown by Hartinger and Milanič in [136] that all 1-perfectly-orientable graphs are $\{\overline{K_2 \cup C_{2k+1}} \mid k \in \mathbb{N}\}$-induced-minor-free. Thus, Corollary 6.2.8 implies that 1-perfectly-orientable graphs form a subclass of $\mathcal{G}_2$. Let us also remark that the proof of the mentioned result in [136] also gives a proof of the fact that nonisomorphic graphs in $\{\overline{K_2 \cup C_{2k+1}} \mid k \in \mathbb{N}\}$ are incomparable with respect to the induced minor relation, which, when combined with Theorem 6.2.6, gives an alternative proof of Corollary 6.2.8.

6.3 k-simplicial elimination orderings and k-simplicial vertices

Recall that a vertex v in a graph G is *k-simplicial* if its neighborhood in G is a union of k cliques. Given a graph G and an integer $k \geq 0$, a *k-simplicial elimination ordering* of G is an ordering $v_1, \ldots, v_n$ of the vertices of G such that for all $i \in \{1, \ldots, n\}$, v_i is k-simplicial in the graph $G[v_1, \ldots, v_i]$. For $k = 1$, resp. $k = 2$, a k-simplicial elimination ordering is also called a *perfect elimination ordering*, resp. a *bisimplicial elimination ordering*.

LexBFS is a linear-time algorithm of Rose, Tarjan, and Lueker [212] whose input is any nonnull graph G together with a vertex $s \in V(G)$, and whose output is an ordering of the vertices of G starting at s. It is a restricted version of Breadth First Search, where the usual queue of vertices is replaced by a queue of unordered subsets of the vertices, which is sometimes refined, but never reordered (for details, see [212]). An ordering of the vertices of a graph G is a *LexBFS ordering* if there exists a vertex s of G such that the ordering can be produced by LexBFS when the input is G, s.

A classical result due to Dirac [95] states that every nonnull chordal graph has a simplicial vertex. An alternative proof of this result was given by Rose, Tarjan, and Lueker [212], who showed that every LexBFS ordering of a chordal graph is a perfect elimination ordering. In this section we generalize these results by showing that for every positive integer k, every LexBFS ordering of a graph in $\mathcal{G}_k$ is a k-simplicial elimination ordering. In particular, this shows that every nonnull graph in $\mathcal{G}_k$ has a k-simplicial vertex. We also examine some algorithmic consequences of these results for the case $k = 2$.

For a family $\mathcal{F}$ of graphs, an ordering $v_1, \ldots, v_n$ of the vertices of a graph G is an *$\mathcal{F}$-elimination ordering* if for every index $i \in \{1, \ldots, n\}$, the graph $G[N_{G[v_1,\ldots,v_i]}(v_i)]$ is $\mathcal{F}$-free. Note that a graph G admits an $\mathcal{F}$-elimination ordering if and only if every nonnull induced subgraph of G contains a vertex whose neighborhood induces an $\mathcal{F}$-free subgraph in G.

In certain cases, $\mathcal{F}$-elimination orderings can be found using LexBFS. This relies on the concept of locally $\mathcal{F}$-decomposable graphs and graph classes, introduced by Aboulker et al. in [2]. Let $\mathcal{F}$ be a family of graphs. A graph G is *locally $\mathcal{F}$-decomposable* if for every vertex v of G, every $F \in \mathcal{F}$ contained, as an induced subgraph, in $G[N_G(v)]$ and every component C of $G \setminus N_G[v]$, there exists $y \in V(F)$ such that y has a non-neighbor in F and has no neighbors in C. A class of graphs $\mathcal{G}$ is *locally $\mathcal{F}$-decomposable* if every graph $G \in \mathcal{G}$ is a locally $\mathcal{F}$-decomposable graph.

Theorem 6.3.1 (Aboulker et al. [2])**.** *If $\mathcal{F}$ is a family of noncomplete graphs and G is a locally $\mathcal{F}$-decomposable graph, then every LexBFS ordering of G is an $\mathcal{F}$-elimination ordering.*

For a hereditary graph class $\mathcal{C}$, we denote by $\mathcal{F}_\mathcal{C}$ the class of all graphs G such that G does not belong to $\mathcal{C}$, but all proper induced subgraphs of G do belong to $\mathcal{C}$.

Theorem 6.3.2. *Let $\mathcal{C}$ be a hereditary class of graphs that is closed under the addition of universal vertices. Then, for every graph $G \in \mathcal{G}_\mathcal{C}$, every LexBFS ordering of G is an $\mathcal{F}_\mathcal{C}$-elimination ordering.*

Proof. First, we note that no graph in $\mathcal{F}_\mathcal{C}$ contains a universal vertex (and thus, $\mathcal{F}_\mathcal{C}$ is a family of noncomplete graphs). Indeed, if some $F \in \mathcal{F}_\mathcal{C}$ contained a universal vertex

u, then the definition of $\mathcal{F}_{\mathcal{C}}$ would imply that $F \setminus u$ belongs to $\mathcal{C}$, and since $\mathcal{C}$ is closed under the addition of universal vertices, it would follow that $F \in \mathcal{C}$, a contradiction.

Now, fix a graph $G \in \mathcal{G}_{\mathcal{C}}$. We claim that G is locally $\mathcal{F}_{\mathcal{C}}$-decomposable. Consider a vertex $x \in V(G)$. Suppose that F is an induced subgraph of $G[N_G(x)]$ such that $F \in \mathcal{F}_{\mathcal{C}}$. By the above, F does not contain a universal vertex, and consequently, every vertex of F has a non-neighbor in F. Let C be a component of $G \setminus N_G[x]$; we must show that some vertex in F is anticomplete to $V(C)$. Suppose otherwise, that is, suppose that every vertex in $V(F)$ has a neighbor in $V(C)$. Let $z \in V(C)$. Clearly, $N_G(x)$ is an (x, z)-separator of G, and moreover, any minimal (x, z)-separator of G included in $N_G(x)$ includes $V(F)$; since $G \in \mathcal{G}_{\mathcal{C}}$ and $\mathcal{C}$ is hereditary, it follows that $F \in \mathcal{C}$, contrary to the fact that $F \in \mathcal{F}_{\mathcal{C}}$. Thus, some vertex in $V(F)$ is indeed anticomplete to $V(C)$. It follows that G is locally $\mathcal{F}_{\mathcal{C}}$-decomposable, and so by Theorem 6.3.1, every LexBFS ordering of G is an $\mathcal{F}_{\mathcal{C}}$-elimination ordering. This completes the proof. □

The reader may wonder whether, in Theorem 6.3.2, it might be possible to eliminate the hypothesis that $\mathcal{C}$ is closed under the addition of universal vertices. This would in fact not be possible (at least not without adding some other, perhaps weaker, hypothesis). To see this, fix any positive integer ℓ, and any hereditary class $\mathcal{C}$ that does not contain K_ℓ. The class $\mathcal{G}_{\mathcal{C}}$ contains all complete graphs, and in particular, $K_{\ell+1} \in \mathcal{G}_{\mathcal{C}}$. However, the neighborhood of any vertex of $K_{\ell+1}$ induces a K_ℓ, and $K_\ell \notin \mathcal{C}$.

Corollary 6.3.3. *Let k be a positive integer. Then, for every graph G in $\mathcal{G}_k$, every LexBFS ordering of G is a k-simplicial elimination ordering.*

Proof. Fix an integer $k \geq 1$. Recall that $\mathcal{C}_k$ is the class of graphs whose vertex set can be partitioned into k cliques. Let $G \in \mathcal{C}_k$, and let $C_1, \ldots, C_k$ be k cliques in G partitioning the vertex set of G. If G' is the graph obtained by adding a universal vertex u to G, then $C_1 \cup \{u\}, C_2, \ldots, C_k$ are k cliques in G' forming a partition of the vertex set of G'. It follows that $\mathcal{C}_k$ is closed under the addition of universal vertices. By Theorem 6.3.2, for every graph G in $\mathcal{G}_k$, every LexBFS ordering of G is a k-simplicial elimination ordering. □

Corollary 6.3.4. *For every positive integer k, every nonnull graph in $\mathcal{G}_k$ has a k-simplicial vertex.*

Remark 6.3.5. Corollary 6.3.3 can also be obtained from the fact that every nonnull graph G contains a *moplex* [24], that is, a clique C such that every two vertices in C have the same closed neighborhood and the neighborhood of C is either empty or a minimal separator of G. (In fact, the last vertex visited by any execution of LexBFS on G necessarily belongs to a moplex.) Given a graph $G \in \mathcal{G}_k$ and a vertex v that belongs to a moplex C of G, there exist k cliques $C_1, \ldots, C_k$ covering the neighborhood of C. But then $C_1, \ldots, C_k \cup (C \setminus \{v\})$ are k cliques covering $N_G(v)$, showing that v is a k-simplicial vertex.

Remark 6.3.6. For $k > 1$, the statement of Corollary 6.3.3 does not generalize to the class of graphs that admit a k-simplicial elimination ordering. To see this, let G be the complete bipartite graph $K_{2,k+1}$. Then, G admits a bisimplicial elimination ordering obtained by placing the two vertices of degree $k+1$ before all the $k+1$ the vertices of degree 2 in the ordering. On the other hand, any LexBFS ordering of G starting at a vertex with degree $k+1$ will end in the other vertex of degree $k+1$, which is not a k-simplicial vertex in G.

6.4 NP-hardness results for $\mathcal{G}_k$, $k \geq 3$

In this section we prove that for all $k \geq 3$, it is NP-hard to recognize graphs in $\mathcal{G}_k$. The time complexity of recognizing graphs in $\mathcal{G}_2$ is still unknown. We also show that the Maximum Clique problem is NP-hard for $\mathcal{G}_3$ (and consequently for $\mathcal{G}_k$ whenever $k \geq 3$).

Recall that for an integer k, the k-Coloring problem is the problem of determining whether the input graph is k-colorable. It is well known that for any integer $k \geq 3$, the k-Coloring problem is NP-complete; we prove the NP-hardness of recognizing graphs in $\mathcal{G}_k$ ($k \geq 3$) by a reduction from this problem. We begin with a technical proposition.

Proposition 6.4.1. *Let $k \geq 0$ be an integer, and let G be a graph. Let G' be the graph obtained from G by adding two new, non-adjacent vertices, and making them adjacent to all vertices of G. Then $\chi(\overline{G}) \leq k$ if and only if $G' \in \mathcal{G}_k$.*

Proof. Let a and b be the two vertices added to G to form G'.

Suppose first that $G' \in \mathcal{G}_k$. Clearly, $V(G)$ is a minimal (a, b)-separator of G', and so $V(G)$ is a union of k cliques of G'. But then $\chi(\overline{G}) \leq k$.

Suppose now that $\chi(\overline{G}) \leq k$; we must show that $G' \in \mathcal{G}_k$. Fix two distinct, non-adjacent vertices $x, y \in V(G')$ and let S be a minimal (x, y)-separator of G'. We must show that S is a union of k cliques of G'.

Suppose first that $\{x, y\} \cap \{a, b\} \neq \emptyset$. By symmetry, we may assume that $x = a$. Since b is the only non-neighbor of a in G', it follows that $y = b$. Since $\{a, b\} = \{x, y\}$ is complete to $V(G) = V(G') \setminus \{a, b\}$, it follows that $V(G)$ is the only (x, y)-separator of G'. So, $S = V(G)$. Since $\chi(\overline{G}) \leq k$, it follows that S is a union of k cliques of G'.

From now on, we assume that $\{x, y\} \cap \{a, b\} = \emptyset$, so that $x, y \in V(G)$. Since x and y are non-adjacent, we see that $V(G)$ is not a clique, and consequently $k \geq \chi(\overline{G}) \geq 2$. Note that $\{a, b\}$ is complete to $\{x, y\}$, and so $a, b \in S$. Now, $\chi(\overline{G}) \leq k$, and so $S \setminus \{a, b\}$ is a union of k cliques of G, say $C_1, \dots, C_k$. Using the fact that $\{a, b\}$ is complete to $V(G)$ in G', and the fact that $k \geq 2$, we see that S is a union of k cliques of G', namely $C_1 \cup \{a\}, C_2 \cup \{b\}, C_3, \dots, C_k$.

We have now shown that $G' \in \mathcal{G}_k$, and we are done. □

Theorem 6.4.2. *For every integer $k \geq 3$, it is NP-hard to recognize graphs in $\mathcal{G}_k$.*

Proof. Fix an integer $k \geq 3$, and let G be any graph. We form a graph G' by adding two new, non-adjacent vertices to $\overline{G}$, and making them adjacent to all vertices of $\overline{G}$. Since $\overline{\overline{G}} = G$, Proposition 6.4.1 guarantees that $\chi(G) \leq k$ if and only if $G' \in \mathcal{G}_k$. Since k-Coloring is NP-complete, it follows that recognizing graphs in $\mathcal{G}_k$ is NP-hard . □

Theorem 6.4.3. *The* Maximum Clique *problem is NP-hard for graphs in $\mathcal{G}_3$.*

Proof. Note that $\mathcal{G}_3$ contains all graphs whose vertex set can be partitioned into three cliques; moreover, note that the vertex set of a graph can be partitioned into three cliques if and only if the complement of the graph is 3-colorable. Thus, it suffices to show that the Maximum Independent Set problem is NP-hard for 3-colorable graphs. But this readily follows from [203]. Indeed, as observed by Poljak [203], for any graph G, the graph G^* obtained from G by subdividing each edge twice has the property that $\alpha(G^*) = \alpha(G) + |E(G)|$. But notice that for any graph G, the graph G^* is 3-colorable. Thus, since the Maximum Independent Set problem is NP-hard for general graphs, it is NP-hard for 3-colorable graphs. □

6.5 Sublasses of $\mathcal{G}_2$

In previous section we saw that the recognition of graphs in $\mathcal{G}_3$ is NP-hard problem. The recognition of graphs in $\mathcal{G}_2$, however, remains an open problem. For this reason, in this section we consider subclasses of the class $\mathcal{G}_2$. In particular, we consider the following subclasses of the class $\mathcal{G}_2$: triangle-free graphs in $\mathcal{G}_2$, perfect graphs in $\mathcal{G}_2$ and diamond-free graphs in $\mathcal{G}_2$.

Recall that a *prism* is any subdivision of $\overline{C_6}$ in which the two triangles remain unsubdivided and *theta* is any subdivision of the complete bipartite graph $K_{2,3}$. In particular, $\overline{C_6}$ is a prism and $K_{2,3}$ is a theta. The diamond, 3-prism and $K_{2,3}$ are depicted in Fig. 6.2. Any prism other than $\overline{C_6}$ is a *long prism*.

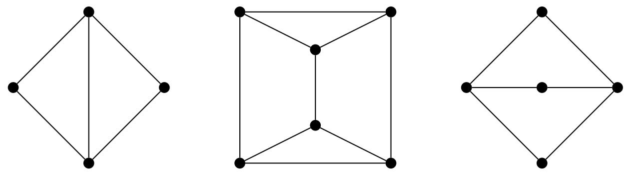


Figure 6.2: From left to right: the diamond, the 3-prism, and the $K_{2,3}$.

A *pyramid* is any subdivision of the complete graph K_4 in which one triangle remains unsubdivided, and of the remaining three edges, at least two edges are subdivided at least once. A 3-path-configuration (or *3PC* for short) is any theta, pyramid, or prism. The three types of 3PC are represented in Fig. 6.3. A *k-prism* is a graph whose vertex set can be partitioned into two k-vertex cliques, say $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_k\}$, such that for all $i, j \in \{1, \dots, k\}$, a_i is adjacent to b_j if and only if $i = j$. Any graph that is a k-prism for some integer $k \geq 3$ will be referred to as short prism. Note that $\overline{C_6}$ is 3-prism.

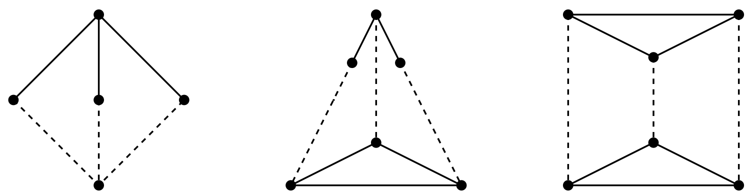


Figure 6.3: Three-path-configurations: theta (left), pyramid (center), and prism (right). A full line represents an edge, and a dashed line represents a path that has at least one edge.

A *broken wheel* (see Fig. 6.4) is a wheel that consists of a hole H and an additional vertex v such that v has at least three neighbors in H, and furthermore, the neighbors of v in $V(H)$ induce a disconnected subgraph of H.

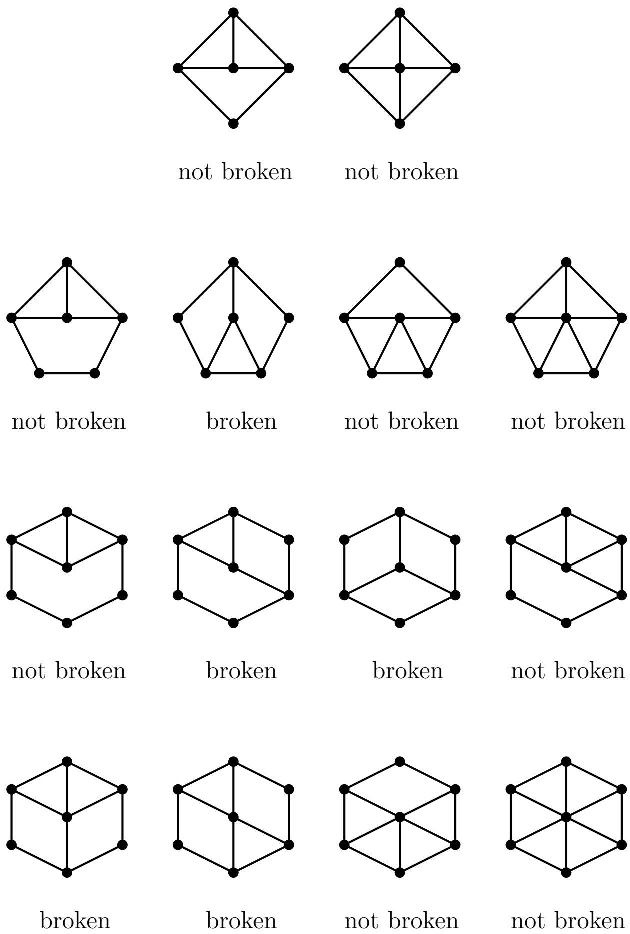


Figure 6.4: Some small wheels, classified as broken or not broken.

6.5.1 Graphs of bounded clique number and perfect graphs in $\mathcal{G}_2$

Among the subclasses of $\mathcal{G}_2$, first we consider the class of graphs of bounded clique number in $\mathcal{G}_2$ and we justify that graphs in this class can be recognized in polynomial time.

Let ℓ be an integer, $\mathcal{G}$ be the class of $K_{\ell+1}$-free graphs and let G be a graph in $\mathcal{G}$, with n vertices and m edges. Now we explain the algorithm that decides whether $G \in \mathcal{G}_2$. We generate the set consisting of all pairs of disjoint cliques in G and enumerate those pairs whose union is a minimal separator in G. Let s_2 be the number of all enumerated pairs; clearly, $s_2 \in \mathcal{O}(n^{2\ell})$. Further, we run some of the algorithms that enumerate all minimal separators of a graph, for example the one developed by Berry et al. [25]. The algorithm enumerates all the minimal separators of G in time $\mathcal{O}(n(n+m)s)$, where s denotes the number of minimal separators of G. We run the algorithm for $\mathcal{O}(n(n+m)s_2)$ time and compare the output with the list of s_2 enumerated clique pairs (more precisely, we compare the output with the sets obtained as a union of cliques from a single pair, for every pair). If the answer is affirmative, we conclude that $G \in \mathcal{G}_2$. Otherwise, if output does not coincide with the list of clique pairs, or the algorithms did not produce any response, we conclude that $G \notin \mathcal{G}_2$.

Now we focus on perfect graphs in $\mathcal{G}_2$. A recent result from Dallard et al. [85]

gives a polynomial-time algorithm for the recognition of graphs that do not contain the complete bipartite graph $K_{2,3}$ as an induced minor.

Theorem 6.5.1 (Dallard et al. [85]). *Determining whether a given graph contains $K_{2,3}$ as an induced minor can be done in time $\mathcal{O}(n^{13}(n+m))$, where n and m denote the number of vertices and edges of the input graph, respectively.*

It turns out that using Theorem 6.5.1 the perfect graphs in $\mathcal{G}_2$ can be recognized in polynomial time. Recall that $\alpha(G)$ is the size of maximum independent set in G, $\omega(G)$ is the size of maximum clique in G, $\theta(G)$ is the minimum number of cliques needed to cover the vertices of G, $\chi(G)$ is the minimum number of colors needed to color the vertices of G so that the adjacent vertices have distinct colors. Note that any independent set in G is a clique in $\overline{G}$, and a proper coloring of G is a clique cover in $\overline{G}$. It follows that for every graph G it holds that $\alpha(\overline{G}) = \omega(G)$ and $\theta(\overline{G}) = \chi(G)$.

Theorem 6.5.2. *Let G be a perfect graph. Then G belongs to $\mathcal{G}_2$ if and only if G is $K_{2,3}$-induced-minor-free.*

Proof. If $G \in \mathcal{G}_2$, then G is $K_{2,3}$-induced-minor free, by Corollary 6.2.2.

Let G be a perfect graph that is $K_{2,3}$-induced-minor-free. Since the complement of a perfect graph is perfect, $\overline{G}$ is perfect as well and by the definition of perfect graphs, for every induced subgraph $\overline{H}$ of $\overline{G}$ it holds that $\omega(\overline{H}) = \chi(\overline{H})$. Note that G is the complement of $\overline{G}$ and the last equality implies that for every induced subgraph H of G it holds that $\alpha(H) = \theta(H)$. Recall that G is $K_{2,3}$-induced-minor free, so every minimal separator S in G induces a graph with independence number at most 2 (see, e.g., [88, Lemma 3.2]). Finally, for an arbitrary minimal separator S in G it holds that $\alpha(G[S]) = \theta(G[S])$, so $\theta(G[S]) \leq 2$. Therefore, S can be covered by two cliques, and since S was an arbitrary minimal separator in G, it follows that $G \in \mathcal{G}_2$. $\square$

The following result is a direct consequence of Theorems 6.5.1 and 6.5.2.

Corollary 6.5.3. *Determining whether a given perfect graph belongs to $\mathcal{G}_2$ can be done in time $\mathcal{O}(n^{13}(n+m))$, where n and m denote the number of vertices and edges of the input graph, respectively.*

6.5.2 Diamond-free graphs in $\mathcal{G}_2$

In this section we prove a decomposition theorem for the class of diamond-free graphs in $\mathcal{G}_2$. The decomposition results imply certain polynomial-time recognition algorithms for these class of graphs, as will be discussed in Section 6.6.2.

We will need the following decomposition theorem for (3PC, wheel)-free graphs.

Theorem 6.5.4 (Conforti et al. [70]). *If a graph G is (3PC, wheel)-free, then either G is a complete graph or a cycle, or G admits a clique cutset.*

Lemma 6.5.5. *$K_{2,3}$ is an induced minor of every theta, pyramid, long prism, or broken wheel.*

Proof. First, we show that $K_{2,3}$ is an induced minor of every theta. Let H be a theta. Let a and b be distinct, non-adjacent vertices of H, and let P^1, P^2, P^3 be distinct induced paths in H, each between a and b, such that any two of P^1, P^2, P^3 have exactly two vertices (namely a and b) in common. Contracting in H all but two edges of each path P^i results in a graph isomorphic to $K_{2,3}$.

Next, we show that $K_{2,3}$ is an induced minor of every pyramid. Let H be a pyramid. Let a be a vertex of H, let $B = \{b_1, b_2, b_3\}$ be a 3-vertex clique in $H \setminus a$, and let P^1, P^2, and P^3 be induced paths in H such that

- for each $i \in \{1, 2, 3\}$, the endpoints of P^i are a and b_i;
- any two of the paths P^1, P^2, P^3 have exactly one vertex (namely a) in common.

Since H is a pyramid, we know that at least two of P^1, P^2, P^3 have more than one edge; by symmetry, we may assume that P^1 and P^2 each have at least two edges. Contracting in H all but two edges of each of the paths P^1 and P^2, all but one edge of the path P^3, and the edge b_1b_2 results in a graph isomorphic to $K_{2,3}$.

Next, we show that $K_{2,3}$ is an induced minor of every long prism. Let H be a long prism. Let $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$ be disjoint 3-vertex cliques in H, and let P^1, P^2, and P^3 be induced paths in H such that

- for each $i \in \{1, 2, 3\}$, the endpoints of P^i are a_i and b_i;
- no two of the paths P^1, P^2, P^3 have any vertices in common.

Since H is a long prism, we know that at least one of P^1, P^2, P^3 has more than one edge; by symmetry, we may assume that P^1 has more than one edge. Contracting in H all but two edges of the path P^1, all but one edge of each of the paths P^2 and P^3, and the edges a_1a_2 and b_1b_3, results in a graph isomorphic to $K_{2,3}$.

Finally, we show that $K_{2,3}$ is an induced minor of every broken wheel. Let W be a broken wheel, consisting of a hole $H = h_0, h_1, \dots, h_{k-1}, h_0$ (with $k \geq 4$, and indices in $\mathbb{Z}_k$) and an additional vertex v that has at least three neighbors in H, and such that the neighbors of v induce a disconnected subgraph of H. By symmetry, we may assume that v is non-adjacent to h_0 and adjacent to h_1. Let the path $h_1, \dots, h_i$ be one component of $H[N_W(v)]$. Contracting in W all edges of the hole H except for the four edges $h_0h_1, h_0h_{k-1}, h_ih_{i+1}, h_{i+1}h_{i+2}$ results in a graph isomorphic to $K_{2,3}$. □

Corollary 6.5.6. *Every graph in $\mathcal{G}_2$ is (theta, pyramid, long prism, broken wheel)-free.*

Proof. Since $\mathcal{G}_2$ is hereditary (by Corollary 6.1.2), it suffices to show that $\mathcal{G}_2$ contains no theta, no pyramid, no long prism, and no broken wheel. By Corollary 6.2.2, $\mathcal{G}_2$ is a subclass of the class of $K_{2,3}$-induced-minor-free graphs. Thus, it suffices to show that the class of $K_{2,3}$-induced-minor-free graphs contains no theta, no pyramid, no long prism, and no broken wheel, or equivalently, that $K_{2,3}$ is an induced minor of every theta, pyramid, long prism, or broken wheel. This is exactly the statement of Lemma 6.5.5. □

Lemma 6.5.7. *Let G be a diamond-free graph that belongs to $\mathcal{G}_2$. Then G is (theta, pyramid, long prism, wheel)-free.*

Proof. Clearly, every wheel either contains an induced diamond or is a broken wheel. The result now follows from Corollary 6.5.6. □

Lemma 6.5.8. *Let G be a (diamond, theta, pyramid, long prism, wheel)-free graph that contains an induced $\overline{C_6}$. Then either G is a short prism, or G admits a clique cutset.*

Proof. Recall that $\overline{C_6}$ is a 3-prism; fix a maximum integer $n \geq 3$ such that G contains an induced n-prism H. Set $V(H) = A \cup B$, where $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$ are disjoint, n-vertex cliques, such that for all $i, j \in \{1, \dots, n\}$, a_i is adjacent to b_j in H if and only if $i = j$. We may assume that $V(H) \subsetneqq V(G)$, for otherwise, G is a short prism, and we are done. We may further assume that G is connected, for otherwise, $\emptyset$ is a clique cutset of G, and again we are done.

Claim 1. For all $v \in V(G) \setminus V(H)$, either $N_G(v) \cap V(H) = A$, or $N_G(v) \cap V(H) = B$, or there exists some $i \in \{1, \dots, n\}$ such that $N_G(v) \cap V(H) \subseteq \{a_i, b_i\}$.

Proof of Claim 1. Fix $v \in V(G) \setminus V(H)$. We may assume that $|N_G(v) \cap V(H)| \geq 2$, for otherwise, the result is immediate. Next, if there exist distinct $i, j \in \{1, \dots, n\}$ such that v is complete to $\{a_i, b_j\}$, then $G[v, a_i, a_j, b_i, b_j]$ is either a theta or a wheel, contrary to the fact that G is (theta, wheel)-free. So, if v has a neighbor both in A and in B, then there exists some $i \in \{1, \dots, n\}$ such that $N_G(v) \cap V(H) = \{a_i, b_i\}$, and we are done. From now on, we assume that either $N_G(v) \cap V(H) \subseteq A$ or $N_G(v) \cap V(H) \subseteq B$; by symmetry, we may assume that $N_G(v) \cap V(H) \subseteq A$, and we deduce that $|N_G(v) \cap A| \geq 2$. Then v is complete to A, for otherwise, we fix pairwise distinct $a_i, a_j, a_k \in A$ such that v is adjacent to a_i, a_j and non-adjacent to a_k, and we observe that $G[v, a_i, a_j, a_k]$ is a diamond, contrary to the fact that G is diamond-free. It now follows that $N_G(v) \cap V(H) = A$, and we are done. This proves Claim 1. ♦

Claim 2. If there exists some $v \in V(G) \setminus V(H)$ such that $N_G(v) \cap V(H) = A$ (resp. such that $N_G(v) \cap V(H) = B$), then A (resp. B) is a clique cutset of G.

Proof of Claim 2. By symmetry, we may assume that some $v \in V(G) \setminus V(H)$ satisfies $N_G(v) \cap V(H) = A$; we must show that A is a clique cutset of G. By construction, A is a clique of G, and so it suffices to show that A is a cutset of G separating v from B. Suppose otherwise. Then there exists an induced path P in $G \setminus V(H)$ between v and some vertex that has a neighbor in B. Let $Q = q_0, \dots, q_t$ (with $t \geq 0$) be a minimum-length subpath of P such that q_0 is complete to A and q_t has a neighbor in B; by Claim 1, $N_G(q_0) \cap V(H) = A$ and $q_0 \neq q_t$, i.e., $t \geq 1$. By symmetry, we may assume that q_t is adjacent to b_1. By Claim 1, we have that either $N_G(q_t) \cap V(H) = B$, or $N_G(q_t) \cap V(H) = \{a_1, b_1\}$, or $N_G(q_t) \cap V(H) = \{b_1\}$.

Assume first that $N_G(q_t) \cap V(H) = B$. If $t = 1$, then $G[A \cup B \cup \{q_0, q_1\}]$ is an $(n+1)$-prism, contrary to the maximality of n. So, $t \geq 2$. By the minimality of Q, all internal vertices of Q are anticomplete to B. If the internal vertices of Q are also anticomplete to A, then $G[\{a_1, a_2, b_1, b_2\} \cup V(Q)]$ is a long prism, contrary to the fact that G is long-prism-free. Hence, some internal vertex of Q has a neighbor in A; let $i \in \{1, \dots, t-1\}$ be maximum with the property that q_i has a neighbor in A. By the minimality of Q, and by Claim 1, we know that q_i has a unique neighbor in A; fix $j \in \{1, \dots, n\}$ such that a_j is the unique neighbor of q_i in A, and fix any $k \in \{1, \dots, n\} \setminus \{j\}$. But now $G[a_j, a_k, b_j, b_k, q_i, q_{i+1}, \dots, q_t]$ is a pyramid, contrary to the fact that G is pyramid-free.

Assume next that $N_G(q_t) \cap V(H) = \{a_1, b_1\}$. Let q_i be the vertex of Q with highest index such that q_i is adjacent to a_2. Then $q_i, \dots, q_t, b_1, b_2, a_2, q_i$ is a hole, and a_1 has at least three neighbors (namely, a_2, q_t, b_1) in it, contrary to the fact that G is wheel-free.

Assume finally that $N_G(q_t) \cap V(H) = \{b_1\}$. Let q_i (resp. q_j) be the vertex of Q with highest index such that q_i (resp. q_j) is adjacent to a_2 (resp. a_1). If $j \geq i$ then $q_i, \dots, q_t, b_1, b_2, a_2, q_i$ is a hole and a_1 has at least three neighbors in it (namely, a_2, q_j, b_1), contrary to the fact that G is wheel-free. So, $i > j$. Then $q_j, \dots, q_t, b_1, a_1, q_j$

is a hole and a_2 has two non-adjacent neighbors in it (namely, a_1, q_i), and hence $G[q_j, \dots, , q_t, b_1, a_1, a_2]$ is a theta or a wheel, contrary to the fact that G is (theta, wheel)-free. This proves Claim 2. ♦

In view of Claims 1 and 2, we assume from now on that for all $v \in V(G) \setminus V(H)$, there exists some $i \in \{1, \dots, n\}$ such that $N_G(v) \cap V(H) \subseteq \{a_i, b_i\}$.

Claim 3. For all $i \in \{1, \dots, n\}$, if some vertex in $V(G) \setminus V(H)$ is complete to $\{a_i, b_i\}$, then $\{a_i, b_i\}$ is a clique cutset of G.

Proof of Claim 3. By symmetry, we may assume that some vertex of $V(G) \setminus V(H)$ is complete to $\{a_1, b_1\}$; we must show that $\{a_1, b_1\}$ is a clique cutset of G. Let p_0 be a vertex of $V(G) \setminus V(H)$ that is complete to $\{a_1, b_1\}$. Since a_1 is adjacent to b_1, it suffices to show that $G \setminus \{a_1, b_1\}$ is disconnected. Suppose otherwise. Then, there exists an induced path in $G \setminus \{a_1, b_1\}$ from the vertex p_0 to a vertex in $V(H) \setminus \{a_1, b_1\}$. Consequently, there exists an induced path $P = p_0, \dots, p_s$ (with $s \geq 0$) in $G \setminus (A \cup B)$ such that p_s has a neighbor in $V(H) \setminus \{a_1, b_1\}$; we may assume that the path P was chosen so that its length is minimum. By Claim 1, we have that $N_G(p_0) \cap V(H) = \{a_1, b_1\}$, and so $s \geq 1$. Furthermore, by the minimality of P, $\{p_0, \dots, p_{s-1}\}$ is anticomplete to $V(H) \setminus \{a_1, b_1\}$. By symmetry, we may assume that $a_2 \in N_G(p_s) \cap V(H) \subseteq \{a_2, b_2\}$. Let i be the largest index in $\{0, \dots, s\}$ such that p_i is adjacent to b_1 (such an i exists because p_0 is adjacent to b_1). Now $p_i, \dots, p_s, a_2, a_3, b_3, b_1, p_i$ is a hole, and a_1 has at least three neighbors (namely, a_2, a_3, b_1) in it, contrary to the fact that G is wheel-free. This proves Claim 3. ♦

In view of Claim 3, we may now assume that no vertex in $V(G) \setminus V(H)$ has more than one neighbor in $V(H)$. Let N_A be the set of all vertices in $V(G) \setminus V(H)$ that have a neighbor in A, and let N_B be the set of all vertices in $V(G) \setminus V(H)$ that have a neighbor in B. Then $N_A \cap N_B = \emptyset$. Since $V(H) \subsetneqq V(G)$ and G is connected, we have that $N_A \cup N_B \neq \emptyset$. If $N_A = \emptyset$, then B is a clique cutset of G, and if $N_B = \emptyset$, then A is a clique cutset of G. So, we may assume that N_A and N_B are both nonempty. Furthermore, we may assume that there is an induced path in $G \setminus V(H)$ between N_A and N_B, for otherwise, both A and B are clique cutsets of G, and we are done. Let $P = p_0, \dots, p_s$ (with $s \geq 0$) be a minimum-length path in $G \setminus V(H)$ such that $p_0 \in N_A$ and $p_s \in N_B$; since $N_A \cap N_B = \emptyset$, we see that $s \geq 1$. Furthermore, the minimality of P implies that the interior of P is anticomplete to $V(H)$.

Since no vertex of $V(G) \setminus V(H)$ has more than one neighbor in $V(H)$, we may assume by symmetry that $N_G(p_0) \cap V(H) = \{a_1\}$, and that either $N_G(p_s) \cap V(H) = \{b_1\}$ or $N_G(p_s) \cap V(H) = \{b_2\}$. But if $N_G(p_s) \cap V(H) = \{b_2\}$, then $G[a_1, a_2, b_1, b_2, p_0, \dots, p_s]$ is a theta, contrary to the fact that G is theta-free. So, $N_G(p_s) \cap V(H) = \{b_1\}$. We now have that $V(P)$ is anticomplete to $V(H) \setminus \{a_1, b_1\}$.

Our goal is to show that $\{a_1, b_1\}$ is a clique cutset of G. Suppose otherwise; then $G \setminus \{a_1, b_1\}$ is connected. Then, there exists an induced path in $G \setminus \{a_1, b_1\}$ from a vertex in P to a vertex in $V(H) \setminus \{a_1, b_1\}$. Since $V(P)$ is anticomplete to $V(H) \setminus \{a_1, b_1\}$, any such path has length at least two. Deleting the endpoints of any such path, we obtain an induced path $Q = q_0, \dots, q_t$ (with $t \geq 0$) in $G \setminus (V(H) \cup V(P))$ such that q_0 has a neighbor in $V(P)$, and q_t has a neighbor in $V(H) \setminus \{a_1, b_1\}$; we may assume that Q is a minimum-length path with this property, so that q_0 is the only vertex of Q with a neighbor in $V(P)$, and q_t is the only vertex of Q with a neighbor in $V(H) \setminus \{a_1, b_1\}$. By symmetry, we may further assume that q_t is adjacent to a_2; then $N_G(q_t) \cap V(H) = \{a_2\}$. Let i be the largest index in $\{0, \dots, s\}$ such that q_0 is adjacent to p_i. Then $p_i, \dots, p_s, b_1, b_3, a_3, a_2, q_t, \dots, q_0, p_i$ is a hole in G, and b_2 has at least three neighbors (namely, a_2, b_1, b_3) in it, contrary to the fact that G is wheel-free. This completes the proof. □

Lemma 6.5.9. *Let G be a (diamond, theta, pyramid, long prism, wheel)-free graph. Then either G is a short prism, a cycle, or a complete graph, or G admits a clique cutset.*

Proof. If G contains an induced $\overline{C_6}$, then the result follows from Lemma 6.5.8. Otherwise, we have that G is (3PC, wheel)-free, and the result follows from Theorem 6.5.4. □

Theorem 6.5.10. *Let G be a diamond-free graph that belongs to $\mathcal{G}_2$. Then either G is a short prism, a cycle, or a complete graph, or G admits a clique cutset.*

Proof. This follows immediately from Lemmas 6.5.7 and 6.5.9. □

A remark about triangle-free graphs in $\mathcal{G}_2$. It turns out that every triangle-free graph G in $\mathcal{G}_2$ is a graph of separability at most 2. *Graphs of separability at most* k, $k \geq 0$, were introduced by Milanič and Cicalese [69] as graphs in which every two non-adjacent vertices in G are separated by a set of at most k other vertices. They showed that graphs of separability at most 2 are precisely the graphs that can be built from complete graphs and cycles by an iterative application of gluing a long 2-cliques. Equivalently, they showed that G is of separability at most 2 if and only if G contains no induced K_5^- (that is, K_5 minus one edge), no induced $3PC$ and no induced wheel. Using this characterization of graphs of separability at most 2 and Corollary 6.5.6 we can easily show that every triangle-free graph in $\mathcal{G}_2$ is of separability at most 2, and thus can be built from complete graphs and cycles by an iterative application of gluing along 2-cliques, by [69, Theorem 1].

6.6 Algorithms and complexity

In this section we use the results and characterizations obtained in previous sections, in order to develop polynomial-time algorithms for certain graph problems in the class $\mathcal{G}_2$ and its subclass of diamond-free graphs in $\mathcal{G}_2$. On the other side, we proved in Section 6.4 that the recognition of graphs in $\mathcal{G}_3$ is NP-hard. Note that, since VERTEX COLORING is NP-hard for circular-arc graphs [116], which form a subclass of $\mathcal{G}_2$, the problem is also NP-hard for $\mathcal{G}_k$, whenever $k \geq 2$. Table 6.1 summarizes our algorithmic and complexity results. Since $\mathcal{G}_0$ is the class of all disjoint unions of complete graphs, and $\mathcal{G}_1$ is the class of all chordal graphs, all problems from the table below can be solved in linear time for $\mathcal{G}_0$ and $\mathcal{G}_1$ (see [114, 212]).

	diamond-free graphs in $\mathcal{G}_2$	$\mathcal{G}_2$	$\mathcal{G}_k$ $(k \geq 3)$
recognition	$\mathcal{O}(n^\omega \log n)$	?	NP-hard
Maximum Weight Clique	$\mathcal{O}(n^\omega \log n)$	$\mathcal{O}(n^{3+o(1)})$	NP-hard
Maximum Weight Independent Set	$\mathcal{O}(n^2(n+m))$	$\mathcal{O}(n^5)$	$\mathcal{O}(n^{2k+2})$
Vertex Coloring	$\mathcal{O}(n^\omega \log n)$	NP-hard	NP-hard

Table 6.1: Summary of our algorithmic and complexity results. The number of vertices and edges of the input graph is denoted by n and m, respectively, and $\omega < 2.3728596$ denotes the matrix multiplication exponent (see [7]).

6.6.1 Algorithmic considerations for graphs in $\mathcal{G}_2$

The Maximum Weight Clique problem can be solved in polynomial time for n-vertex graphs that admit a bisimplicial elimination ordering, see [233]. The algorithm iteratively removes bisimplicial vertices and reduces the problem to solving n instances of the Maximum Weight Independent Set problem in bipartite graphs. The polynomial running time of the algorithm given in [233] was based on polynomial-time solvability of the Maximum Weight Independent Set problem in the class of perfect graphs. Using maximum flow techniques, an improved running time of $\mathcal{O}(n^4)$ can be achieved, see [17]. For graphs in $\mathcal{G}_2$, a further improvement can be obtained using LexBFS and recent developments on maximum flow algorithms.

Theorem 6.6.1. *For every $\epsilon > 0$, the* Maximum Weight Clique *problem can be solved in $\mathcal{O}(n^{3+\epsilon})$ time for n-vertex graphs in $\mathcal{G}_2$.*

Proof. Let G be an n-vertex graph in $\mathcal{G}_2$. In time $\mathcal{O}(n^2)$, we compute a LexBFS ordering $v_1, \ldots, v_n$ of G. By Corollary 6.3.3, $v_1, \ldots, v_n$ is a bisimplicial elimination ordering of G. For each $i \in \{1, \ldots, n\}$, let G_i be the graph induced by the closed neighborhood of v_i in the graph $G[v_1, \ldots, v_i]$. We will show that for each $i \in \{1, \ldots, n\}$, we can compute a maximum-weight clique C_i in G_i in time $\mathcal{O}(n^{2+\epsilon})$. This will suffice, since a clique C_i with maximum total weight is also a clique in G with maximum total weight.

Fix an $i \in \{1, \ldots, n\}$. A maximum-weight clique in the graph G_i consists of v_i and a maximum-weight independent set in the complement of the graph $G_i - v_i$. Since v_i is a bisimplicial vertex in the graph $G[v_1, \ldots, v_i]$, the complement of the graph $G_i - v_i$ is bipartite. The problem of finding a maximum-weight independent set in a vertex-weighted bipartite graph can be reduced in linear time to a maximum flow problem in a derived network (see, e.g., [141]). Using a recent result due to Chen et al. [57] showing that the maximum flow problem can be solved in almost linear time, we conclude that a maximum-weight independent set in the complement of the graph $G_i - v_i$ can be computed in time $\mathcal{O}(n^{2+\epsilon})$. The claimed $\mathcal{O}(n^{3+\epsilon})$ overall time complexity follows. □

Note that Theorem 6.4.3 implies that this result cannot be generalized to graphs in $\mathcal{G}_k$ for $k \geq 3$, unless P = NP.

Dallard et al. showed in [88] that, for each positive integer k, the Maximum Weight Independent Set problem can be solved in $\mathcal{O}(n^{2k})$ time for n-vertex $K_{2,k}$-induced-minor-free graphs. To connect this result with the classes $\mathcal{G}_k$, recall that every graph in $\mathcal{G}_k$ is $K_{2,k+1}$-induced-minor-free, by Corollary 6.2.2. Therefore, the result by Dallard et al. implies the following.

Corollary 6.6.2. *For each integer $k \geq 0$, the* Maximum Weight Independent Set *problem can be solved in $\mathcal{O}(n^{2k+2})$ time for n-vertex graphs in $\mathcal{G}_k$.*

Similar results hold for a number of other related problems, including the Maximum Induced Matching problem, the Dissociation Set problem, etc. We refer to [87] for the details.

Furthermore, Dallard et al. showed in [86] that in any class of $K_{2,k}$-induced-minor-free graphs, the treewidth of the graphs in the class is bounded from above by some polynomial function of the clique number (see also [88]). Combining this with a result of Chaplick et al. [56, Theorem 12], it follows that for any two positive integers k and ℓ, the ℓ-Coloring problem is solvable in time $\mathcal{O}(n)$ in the class of n-vertex $K_{2,k}$-induced-minor-free graphs, and thus in the class $\mathcal{G}_k$ as well. (The $\mathcal{O}$-notation hides a constant depending on k and ℓ.) The same result holds in fact for the more general List ℓ-Coloring problem, in which every vertex is equipped with a list of available colors from the set $\{1, \ldots, \ell\}$.

6.6.2 Algorithmic considerations for diamond-free graphs in $\mathcal{G}_2$

Clearly, short prisms, cycles, and complete graphs are diamond-free and belong to $\mathcal{G}_2$, and furthermore, they can all be recognized in polynomial time. So, using Corollary 6.1.4 and Theorem 6.5.10, we show that diamond-free graphs in $\mathcal{G}_2$ can be recognized in polynomial time. In order to derive the result, we decompose a graph by means of clique cutsets. This common algorithmic tool, first proposed by Tarjan (see [226]), applies to any n-vertex graph G and produces a family $\mathcal{H}$ of $\mathcal{O}(n)$ induced subgraphs of G that do not have any clique cutsets and such that G can be obtained by an iterative application of gluing graphs from $\mathcal{H}$ along cliques. The original algorithm proposed by Tarjan runs in time $\mathcal{O}(n(n+m))$, where m denotes the number of edges of G. A more efficient approach for decomposing a graph along clique cutsets was suggested by Coudert and Ducoffe [75]. They improved the time complexity to $\mathcal{O}(n^\omega \log n)$, where $\omega < 2.3728596$ is the matrix multiplication exponent (see [7]).

Proposition 6.6.3. *There exists an algorithm running in time $\mathcal{O}(n^\omega \log n)$ that correctly determines if an input n-vertex graph G is a diamond-free graph in $\mathcal{G}_2$.*

Proof. Given a graph G with n vertices, testing if G is diamond-free can be done in time $\mathcal{O}(n^\omega)$ (see [231]). Assuming G is diamond-free, we compute the connected components of G and run the algorithm by Coudert and Ducoffe [75] on each nontrivial component of G. This can be done in time $\mathcal{O}(n^\omega \log n)$. The algorithm produces a family $\mathcal{H}$ of $\mathcal{O}(n)$ induced subgraphs of G that do not have any clique cutsets and such that G can be obtained by an iterative application of gluing graphs from $\mathcal{H}$ along cliques. We then check, for each graph $H \in \mathcal{H}$, whether H is a short prism, cycle, or a complete graph. If this is the case, the algorithm determines that G belongs to $\mathcal{G}_2$, and otherwise, it determines that G does not belong to $\mathcal{G}_2$. The correctness follows from Corollary 6.1.4 and Theorem 6.5.10.

To complete the proof, we show that testing whether a given $H \in \mathcal{H}$ satisfies one of the desired properties can be done in time $\mathcal{O}(n+m)$, where m denotes the number of edges of G. Since H is connected, testing if it is a cycle or a complete graph can be done in linear time simply by checking if all the vertex degrees are equal to 2 or to $|V(H)| - 1$, respectively. If none of these cases occurs, we can assume that $n = 2k$ for some $k \geq 3$ and that every vertex in H has degree exactly k, since otherwise, we can infer that H is not a short prism. We choose an arbitrary vertex $v \in V(H)$ and compute the components of the graph $H[N_H(v)]$. If H is a k-prism, then $H[N_H(v)]$ has exactly two components, say C and D, such that C is isomorphic to a complete graph K_{k-1} and D is a trivial component. Set $A = C \cup \{v\}$ and $B = V(G) \setminus A$. Since we already checked that all the vertices are of degree $k+1$, it remains to verify if B is a clique of cardinality k. If this is the case, then H is a short prism, otherwise it is not. Each of the above constantly many steps can be carried out in linear time. □

Moreover, it is clear that the Maximum Weight Clique, Maximum Weight Independent Set, and Vertex Coloring can be solved in polynomial time for short prisms, cycles, and complete graphs. Thus, Theorem 6.5.10 and the algorithm by Coudert and Ducoffe [75] allow us to solve these three optimization problems in polynomial time for diamond-free graphs in $\mathcal{G}_2$. A more precise time complexity analysis is provided by the following.

Theorem 6.6.4. *When restricted to the class of diamond-free graphs in $\mathcal{G}_2$ with n vertices and m edges, the* Maximum Weight Clique *and* Vertex Coloring *problems can be solved in $\mathcal{O}(n^\omega \log n)$ time and the* Maximum Weight Independent Set *problem in $\mathcal{O}(n^2(n+m))$ time.*

Proof. Let G be a diamond-free graph with n vertices and m edges that belongs to $\mathcal{G}_2$. For the Maximum Weight Clique and Vertex Coloring problems, the approach is as follows. We compute the connected components of G and run the algorithm by Coudert and Ducoffe [75] on each component of G. This can be done in time $\mathcal{O}(n^\omega \log n)$. We obtain a family $\mathcal{H}$ of $\mathcal{O}(n)$ induced subgraphs of G that do not have any clique cutsets and such that G can be obtained by an iterative application of gluing graphs from $\mathcal{H}$ along cliques. We now iterate over all $H \in \mathcal{H}$ and solve the Maximum Weight Clique and Vertex Coloring problems on H in linear time. By Theorem 6.5.10, each graph $H \in \mathcal{H}$ is a short prism, cycle, or a complete graph; as explained in the proof of Proposition 6.6.3, which of these cases occurs can be determined in linear time in the size of H. If H is a complete graph, then its chromatic number is $|V(H)|$ and its vertex set solves the Maximum Weight Clique problem. Otherwise, if H is a cycle with at least four vertices, then its chromatic number is either 2 or 3, depending on whether $|V(H)|$ is even or odd, respectively, and to solve the Maximum Weight Clique problem, we only need to examine its edges. Finally, if H is a short prism, say with $|V(H)| = 2k$ for some $k \geq 3$, then we can identify in linear time the two cliques A and B, each of size k, that partition $V(H)$. Then, the chromatic number of H is k, and to solve the Maximum Weight Clique problem, we only need to examine the two cliques A and B and the k edges connecting them. In all these cases, the Maximum Weight Clique and Vertex Coloring problems can be solved in linear time for graphs in $\mathcal{H}$. Since each clique of G is fully contained in one of the graphs in $\mathcal{H}$, this provides an efficient solution to the Maximum Weight Clique problem on G. Similarly, the chromatic number of G is the maximum chromatic number of the graphs in $\mathcal{H}$.

For the Maximum Weight Independent Set problem, the approach is similar, except that in each decomposition step decomposes G along a cut-partition (A, B, C) of G such that C is a clique and the subgraph of G induced by $A \cup C$ belongs to $\mathcal{H}$. For each vertex $v \in C$, we determine a maximum-weight independent set of the subgraph of G induced by $A \setminus N_G(v)$, a maximum-weight independent set of the subgraph of G induced by A, redefine the weights on C, and solve the problem recursively on the graph $G - A$. (We refer to [226] for details; see also [36, Section 8.1].) Thus, we solve $\mathcal{O}(|V(G)|)$ subproblems for each graph $H \in \mathcal{H}$ for a total of $\mathcal{O}(|V(G)|^2)$ subproblems. Each of these subproblems can be solved in linear time. If H is a complete graph, then a heaviest vertex forms a maximum-weight independent set. If H is a cycle, then we can use the fact that cycles have bounded treewidth and apply the results from [11, 31]. Finally, assume that H is a short prism. Then, for each vertex v a maximum-weight independent set S_v containing v can be computed in $\mathcal{O}(|V(H)|)$ time: indeed, S_v is of the form $S_v = \{v, z^v\}$ where z^v is a vertex of maximum weight among the non-neighbors of v in H. The heaviest among the sets S_v forms a maximum-weight independent set. The complexity of this approach is $\mathcal{O}(|V(H)|^2)$, which in this case is $\mathcal{O}(|V(H)| + |E(H)|)$. $\square$

Chapter 7

Final Remarks to Part I

Some of the results we discussed earlier in Part I have been previously published in the proceedings of peer-reviewed conferences and papers in scientific journals. Several of these findings have sparked interest among other researchers in structural and algorithmic graph theory, leading to further studies of the problems studied in our work. In this chapter, we summarize the results from the recent literature motivated by our works and discuss any remaining unanswered questions from our research, as well as potential future research directions.

7.1 Tame graph classes

In Chapter 4 we considered tame graph classes, that is, classes of graphs with polynomially many minimal separators, and examined them from various points of view. Constructions of graphs with exponentially many minimal separators and the newly identified families of tame graph classes led to a complete classification of the tame graph classes within the family of graph classes defined by forbidden induced subgraphs with at most four vertices (see Section 4.4). These results were published in paper [183]), and were followed by research papers on the same topic by other authors, as follows.

In parallel with our studies, Chudnovsky et al. [66] gave a polynomial-time algorithm to compute a maximum weight independent set in an (even-hole, pyramid)-free graph. They obtained a decomposition result for the class of (even-hole, pyramid)-free graphs and using it, they proved that this class of graphs is tame. As a continuation of these studies, Abrishami et al. confirmed a conjecture posed in [66]. In particular, they showed that the class of (theta, pyramid, prism, turtle)-free graphs is tame. Furthermore, they conjectured that for every k the class of all k-creature-free graphs is tame.

Soon after, a systematic study of tame graph classes was done by Gartland and Lokshtanov [120, 121]. They say that a graph class is *strongly-quasi-tame* if there exists a constant c such that every graph in the class on n vertices contains at most $\mathcal{O}(n^c \log n)$ minimal separators. Recall that a graph class is *feral* if there exists a constant $c > 1$ such that for arbitrarily large n there is an n-vertex graph in the class with at least c^n minimal separators. While for any tame graph class it clearly holds that it is not feral, the opposite is in general not true: there are graph classes that are neither feral nor tame, see [115]. Gartland and Lokshtanov gave a counterexample to the conjecture of Abrishami et al. [3] and obtained a complete classification of all hereditary graph classes defined by a finite set of forbidden induced subgraphs into strongly-quasi-tame or feral. In their work the weaker forms of the conjecture by Abrishami et al. were proved. In particular, they showed that for every k the class of all k-creature-free graphs that additionally excludes all cycles of length at least ℓ, for some constant ℓ, is tame. Furthermore, they showed that for every k the class of all k-creature-free graphs that is K_ℓ-free for some fixed constant ℓ is tame. For details, see the paper by Gartland and Lokshtanov [120] (as well as [118]).

Gartland and Lokshtanov obtained a complete classification of graph families defined by a finite number of forbidden induced subgraphs into strongly-quasi-tame and feral and in particular, they showed that the absence of k-creatures, when combined with an excluded k-skinny ladder[1] as an induced minor, implies a quasi-polynomial upper bound on the number of minimal separators. They conjectured that the resulting graph classes are tame and showed that proving their tameness would imply a full dichotomy of hereditary graph classes defined by a finite set of forbidden induced subgraphs into tame and feral. Their conjecture was proved by Gajarský et al. [115] in the form of the following result (Theorem 4.5.13): for every positive integer k, the class of graphs that are k-creature-free and do not contain a k-skinny-ladder as an induced minor is tame. Finally, this result was followed by the second paper of Gartland and Lokshtanov where they showed that every graph class that is (k-creature, k-critter)-free is quasi-tame (for the definition of k-critter and other details, see [121]).

As proved by Gartland and Lokshtanov in [120], Theorem 4.5.13, combined with [120, Theorem 1.3], implies a dichotomy theorem classifying graph classes defined by finitely many induced subgraphs into tame and feral. This result implies that graph class defined by a *single* forbidden induced subgraph F is tame if F is an induced subgraph of the path P_4 or of the graph $2P_2$, and feral, otherwise. We address the same question for the induced minor and induced topological minor relations. Building on the result of Gajarský et al. [115], we show that the class consisting of graphs that do not contain any induced subdivision of the house is tame. Together with the result that the class of butterfly-induced-minor-free graphs is tame, this leads to a characterization of tame graph classes defined by a single forbidden induced minor, or induced topological minor.

A natural research question that arises here is a characterization of tame graph classes by a single forbidden minor or topological minor. Furthermore, our studies can be extended to a more general study classifying tame graph classes by finitely many forbidden induced minors, or finitely many forbidden induced topological minors.

7.2 Extremal number of minimal separators

In Chapter 5 we studied several tame graph classes satisfying the property that the number of minimal separators of every graph in a class is bounded by the number of vertices in a graph. This leads to the general research problem of identification of all graph classes for which the number of minimal separators of every graph in a class is bounded by the number of vertices. In other words, we wonder whether one can characterize tame graph classes for which the polynomial that bounds the number of minimal separators is the identity function.

[1] A k-skinny ladder is a graph consisting of two induced anti-adjacent paths $P = (p_1, \ldots, p_k)$, $Q = (q_1, \ldots, q_k)$, an independent set $R = (r_1, \ldots, r_k)$, and edges $\bigcup_{i=1}^{k} \{p_i r_i, q_i r_i\}$ (see Section 4.5.2).

In particular, we say that a graph G is *orderly* if $s(G) \leq |V(G)|$ and a graph class is *orderly* if every graph in the class is orderly. The proposed research question is the classification of orderly graph classes, that is, the study of sufficient and necessary conditions for a graph class to be orderly. These are the vertices, the removal of which reduces the number of minimal separators in a graph by at most one.

A vertex v in a graph G is said to be *well-behaved* if its deletion decreases the number of minimal separators by at most 1, that is, if $s(G-v) \geq s(G)-1$. Note that in this case $s(G-v) \leq s(G) \leq s(G-v)+1$. We say that a graph G is *well-behaved* if every induced subgraph of G contains a well-behaved vertex (see Fig. 7.1). Not all graphs are well-behaved. For example, if G is the cycle C_n for $n \geq 3$, then G has $\binom{n}{2} - n$ minimal separators, while the deletion of any vertex results in the path P_{n-1}, which has $n-3$ minimal separators; the difference $\binom{n}{2} - 2n+3$ exceeds 1 for all $n \geq 5$. The importance of well-behaved graphs for the study of minimal separators is evident from the following.

Proposition 7.2.1. *Any class of well-behaved graphs is orderly. More precisely, if $\mathcal{G}$ is a class of well-behaved graphs, then $f_{\mathcal{G}}(n) \leq n-1$ for all $n \in \mathbb{N}$.*

Proof. Let $\mathcal{G}$ be a class of well-behaved graphs and let $\mathcal{H}$ be the class of all induced subgraphs of graphs in $\mathcal{G}$. Since $\mathcal{G}$ is a subclass of $\mathcal{H}$, it suffices to show that $f_{\mathcal{H}}(n) \leq n-1$ for all $n \in \mathbb{N}$.

We prove the inequality by induction on n. If $n=1$, then the only possible n-vertex graph in $\mathcal{H}$ is P_1 and $s(P_1)=0$, hence $f_{\mathcal{H}}(1) \leq 0$. Assume now that $n>1$ and let H be an n-vertex graph in $\mathcal{H}$. By the definition of $\mathcal{H}$, there exists a graph $G \in \mathcal{G}$ such that H is an induced subgraph of G. Since G belongs to $\mathcal{G}$, it is well-behaved, hence the induced subgraph H of G contains a well-behaved vertex v. It follows that $s(H-v) \geq s(H)-1$. Since the graph $H-v$ is an induced subgraph of G, it belongs to $\mathcal{H}$. By the induction hypothesis, we have $f_{\mathcal{H}}(n-1) \leq n-2$, which implies $s(H-v) \leq n-2$ and consequently $s(H) \leq s(H-v)+1 \leq n-1$. It follows that $f_{\mathcal{H}}(n) \leq n-1$, as claimed. □

As we show next, many well-known properties of vertices of graphs guarantee that the vertex is well-behaved.

Proposition 7.2.2. *For every graph G it holds that every vertex that is universal, or simplicial, or has a true twin, or has a false twin, or has codegree 1, is 1-well-behaved.*

Proof. Let G be a graph and let v be a vertex in G. If v is universal, simplicial, or has a true or a false twin, then the claim follows from Propositions 3.2.11 and 3.2.13 to 3.2.15, respectively.

Let v be a vertex of codegree 1 in G. Let u be the unique vertex of $V(G) \setminus \{v\}$ that is non-adjacent to v. Let $G' = G-v$. Let S be a minimal separator in G. If $v \in S$, then $S \setminus \{v\}$ is a minimal separator in G', by Corollary 3.2.8. If $v \notin S$, then $v \in A \cup B$, where A and B are S-full components of $G-S$, since otherwise v would have a non-neighbor in each of A and B, contradicting the assumption that u is the unique non-neighbor of v in G. We may assume without loss of generality that $v \in A$. Thus $V(B) = \{u\}$, since A and B are anticomplete to each other. We know that $N_{G'}(u) = N_G(u) \subseteq N_{G'}(v)$, so $S = N_G(u)$. □

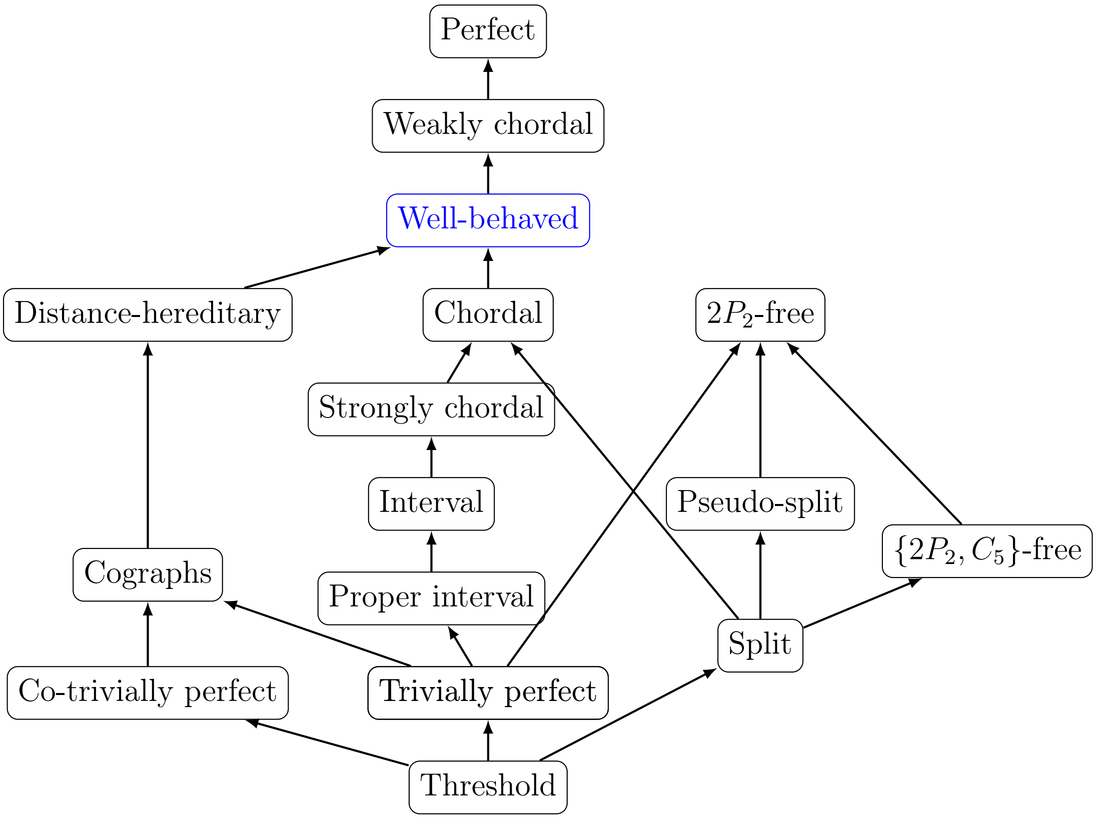


Figure 7.1: Graph classes studied in Chapter 5 and the class of well-behaved graphs. An arrow from a class $\mathcal{G}_1$ to a class $\mathcal{G}_2$ means that every graph in $\mathcal{G}_1$ is also in $\mathcal{G}_2$.

The result about codegree one is sharp in the following sense: we cannot increase the codegree to beyond one, since a vertex of codegree two is not necessarily well-behaved. For example, every vertex of the 5-cycle has codegree two, but deleting any such vertex reduces the number of minimal separators by 3. The same graph shows that bisimplicial vertices are not well-behaved. One more example of vertex that is not well-behaved is a cut-vertex, as can be seen by gluing two cycles along a vertex. The following observation describes a class that is not well-behaved.

Observation 7.2.3. *Complements of cycles are not well-behaved graphs.*

Proof. Observe first that $s(\overline{C_n}) = n$ for $n \geq 5$. Since $s(\overline{P_1}) = 0$ and the graph $\overline{P_n}$ (for any $n \geq 2$) can be obtained from the graph $\overline{P_{n-1}}$ by adding to it a vertex of codegree one, we infer that $s(\overline{P_n}) \leq n-1$ for all $n \geq 2$. Deleting any vertex from $\overline{C_n}$ results in the graph $\overline{P_{n-1}}$, and $s(\overline{P_{n-1}}) \leq n-2$. □

Consequently, every well-behaved graph is weakly chordal (see Fig. 7.1). This inclusion is proper, as can be seen by domino, the graph obtained by gluing two 4-cycles along a 2-clique. Furthermore, every well-behaved graph is orderly as well, while orderly graph classes do not necessarily consist of well-behaved graphs only. For example, the class of all complements of cycles is orderly (every graph in the class is $2P_2$-free and so Theorem 5.3.6 applies), even though it is not a well-behaved class of graphs.

This study can be extended in several different directions. Here we propose some open problems for future research.

(1) Characterize the class of well-behaved graphs by minimal forbidden induced subgraphs.

(2) Let $\mathcal{G}$ be a class of graphs G such that every induced subgraph H of G satisfies that $s(H) \leq |V(H)|$. Characterize $\mathcal{G}$ by minimal forbidden induced subgraphs.

(3) Determine the extremal number of minimal separators for other tame graph classes. Consider, for example, the class of *ladders* (Cartesian products of P_2 and a path), that is an orderly class of graphs. Then every subclass of this class is orderly as well, and we wonder what is the extremal number of minimal separators in such graph classes. More generally, we may consider the class of bipartite permutation graphs, biconvex bipartite graphs, convex bipartite graphs, chordal bipartite graphs, and weakly chordal graphs.

7.3 Bisimplicial separators

In Chapter 6 we considered graphs in which every minimal separator is k-simplicial, that is, a union of k cliques, for some fixed integer $k \geq 0$. We denoted by $\mathcal{G}_k$ the graph class consisting of graphs with k-simplicial minimal separators and showed that the recognition of graphs in $\mathcal{G}_k$ is NP-hard for every $k \geq 3$. As the cases $k = 0$ and $k = 1$ correspond to classes of complete graphs and chordal graphs, respectively, the only remaining case where the recognition problem is open is $\mathcal{G}_2$. This is the main open question with respect to the study of k-simplicial graphs.

Question 7.3.1. Can graphs in $\mathcal{G}_2$ be recognized in polynomial time?

Regarding subclasses of $\mathcal{G}_2$, in Chapter 6 we considered the intersection of $\mathcal{G}_2$ with classes of graphs of bounded clique number, the class of perfect graphs, and the class of diamond-free graphs, and proposed polynomial-time algorithms for the recognition of graphs from these classes as well. It turns out that wheel-free graphs in $\mathcal{G}_2$ have the same structural characterization as diamond-free graphs in $\mathcal{G}_2$, which yields a polynomial-time recognition algorithm for this class of graphs, too. Moreover, since any graph in $\mathcal{G}_2$ is (theta, pyramid, long prism, broken wheel)-free, the class of wheel-free graphs in $\mathcal{G}_2$ coincides with the class of consecutive-wheel-free graphs in $\mathcal{G}_2$, where a *consecutive wheel* is a wheel that consists of a hole H and a center v such that the neighborhood of v in $V(H)$ induces a connected subgraph of H. Every wheel is either broken or consecutive. One can show that every consecutive-wheel-free graph in $\mathcal{G}_2$ is either a short prism, a cycle, or a complete graph, or admits a clique-cutset. The proof of this structural result is quite similar to the one presented for diamond-free graphs, so we omit it here.

Regarding the overall study of the class $\mathcal{G}_2$, we point out the following remarks. First, note that the subclass of the class $\mathcal{G}_2$ that consists of prism-free graphs is contained in the class of (theta, pyramid, prism, turtle)-free graphs and is tame by [3] and one can decide in polynomial time whether such a graph belongs to the class $\mathcal{G}_2$. Thereby, the problem of recognizing graphs in $\mathcal{G}_2$ can be reduced to the subclass of $\mathcal{G}_2$ consisting of graphs that contain an induced short prism and an induced consecutive wheel.

Question 7.3.2. Can we find a decomposition theorem for (theta, pyramid, long prism, broken wheel)-free graphs that contain an induced short prism, and an induced consecutive wheel?

Let us now put the complexity results in the class of diamond-free graphs in perspective by comparing them with the known complexities of the three problems in the larger classes of diamond-free graphs and graphs in $\mathcal{G}_2$. First, the Vertex Coloring problem is NP-hard for diamond-free graphs [159], as well as for graphs in $\mathcal{G}_2$, since it is already hard for the subclass of circular-arc graphs [116]. The situation is somewhat different for the Maximum Weight Independent Set problem, which is NP-hard (even in the unweighted case) in the class of diamond-free graphs, as can be seen using Poljak's reduction [203], but solvable in polynomial time in the class $\mathcal{G}_2$. Our algorithm described in Corollary 6.6.2 solves the problem in $\mathcal{G}_2$ in time $\mathcal{O}(n^6)$. Recently, Dallard, Milanič and Štorgel improved this result and obtained the algorithm that solves the same problem for n-vertex $K_{2,3}$-induced-minor-free graph in time $\mathcal{O}(n^5)$ (see [88, Theorem 3.16]).

Finally, while the Maximum Weight Clique problem is known to be solvable in polynomial time both for diamond-free graphs as well as for graphs in $\mathcal{G}_2$, the running time of the algorithm given by Theorem 6.6.4 improves on both time complexities. By Theorem 6.6.1, the problem can be solved in $\mathcal{O}(n^{3+\epsilon})$ time for n-vertex graphs in $\mathcal{G}_2$, for every $\epsilon > 0$. For the class of diamond-free graphs, observe that every edge in such a graph is contained in a unique maximal clique. Thus, a diamond-free graph with n vertices and m edges has $\mathcal{O}(n+m)$ maximal cliques, and the Maximum Weight Clique problem can be solved in polynomial time by enumerating all maximal cliques and returning one of maximum weight. Using, for example, the maximal clique enumeration algorithm due to Makino and Uno [172], this would result in an overall running time of $\mathcal{O}(n^{2.373}(n+m))$ on diamond-free graphs with n vertices and m edges.

Part II

Part II: Independent sets

Chapter 8

Overview

An *independent set* in a graph G is a set of pairwise non-adjacent vertices in G. An independent set is said to be *maximum* if it has maximum cardinality among all independent sets in G and *maximal* if it is not contained in any larger independent set. Given a graph G and an integer k, deciding whether G contains an independent set of cardinality k is an NP-complete problem known under the name Independent Set [146].

If every vertex of a graph G is assigned a real number, the *weight* of a vertex, we speak about a *weighted graph*. Independent Set naturally generalizes to a weighted version. Maximum Weight Independent Set (MWIS) is the problem of computing an independent set of maximum weight in a given weighted graph, where the weight of a set of vertices is defined as the sum of the weights of its members.

It is known that Independent Set remains NP-hard on $\mathcal{F}$-free graphs (where $\mathcal{F}$ is a finite set) unless, for at least one graph in $\mathcal{F}$, every connected component is a path or a subdivision of the claw [5].

In this part of the thesis we consider two distinct problems related to the independent set problem. The first one is the problem of computing the vector space consisting of all vertex weight functions under which all the maximal independent sets of the graph have constant weight. The second one is an allocation problem, a problem of computing an optimal allocation of items to agents in the presence of a conflict graph, respecting a certain fairness criterion.

8.1 Well-Covered Vector Spaces

While every maximum independent set in a graph is also a maximal one, the opposite implication does not hold. If every maximal independent set in a graph G is also a maximum one, the graph G is said to be *well-covered*. Well-covered graphs were introduced by Plummer in 1970 [201] and have been extensively studied in the literature (see [137] for an introduction and [202] for a survey). One of the motivations for the study of well-covered graphs stems from the fact that Maximum Independent Set is solvable in linear time in the class of well-covered graphs by a simple greedy algorithm that computes a maximal independent set.

Two central research directions on well-covered graphs are the study of their recognition and their characterizations in special graph classes. As proved independently by Sankaranarayana and Stewart in 1992 [214] and by Chvátal and Slater in 1993 [68], the recognition of well-covered graphs is co-NP-complete. In Plummer's survey from 1993 (see [202]) one can find results on various restrictions of the well-coveredeness property defining special subclasses of well-covered graphs, as well as an overview of the study of well-coveredeness versus the girth and the maximum degree. After Plummer's survey, the study of well-covered graphs focused mostly on the recognition problem in special cases. In particular, Caro, Sebő, and Tarsi showed that the recognition of well-covered graphs remains co-NP-complete even for $K_{1,4}$-free graphs [53], Brown and Hoshino established co-NP-completeness for the class of circulant graphs [49], and a careful examination of the reduction due to Sankaranarayana and Stewart [214] shows that the problem remains co-NP-complete in the class of weakly chordal graphs. On the positive side, Tankus and Tarsi showed that the problem is polynomial-time solvable in the class of claw-free graphs (see [224, 225]). The well-coveredness property can also be tested efficiently in the classes of bipartite graphs [103, 211], graphs with girth at least 5 [106], graphs without cycles of lengths 4 and 5 [107], chordal graphs [206], graphs of bounded degree [52], perfect graphs with bounded clique number [91], various generalizations of the class of cographs [9, 147], and graphs of bounded clique-width [8]. The problem has also been studied from the parameterized complexity point of view, by Alves et al. [8] and Araújo et al. [9].

In this thesis we focus on a weighted generalization of well-coveredness. Given a graph G and a weight function $w : V(G) \to \mathbb{R}$, a graph G is said to be *w-well-covered* if all maximal independent sets in G are of the same weight with respect to the weight function w. The concept of w-well-covered graphs was introduced by Caro, Ellingham, and Ramey in 1998 [52], in the more general context of weight functions mapping the vertices of a graph to the elements of an abelian group (see also [50]). Graphs that are w-well-covered with respect to some *nonnegative* weight function $w : V(G) \to \mathbb{R}_+$ that is not identically equal to 0 are exactly the complements of the so-called *stochastic* graphs studied in 1983 by Berge [21], and generalize the *equistable* graphs introduced in 1980 by Payan [194] and defined as the graphs that admit a weight function $w : V(G) \to \mathbb{R}_+$ such that a set $S \subseteq V(G)$ is a maximal independent set if and only if the total weight of S equals 1.

Given a graph G, a *well-covered weighting* of G is any real-valued weight function w on the vertices of G such that G is w-well-covered. For every graph G, the set $\mathrm{WCW}(G)$ of all well-covered weightings of G forms a vector space over the field of real numbers (see [50, 54]); we refer to it as the *well-covered vector space* of G. Similar vector spaces can be defined for more general situations, for example for hypergraphs and for vertex weight functions that assign to each vertex of G a value from some fixed field $\mathbb{F}$ (see Caro and Yuster [54] and Brown and Nowakowski [50]). In this thesis we restrict our attention to the case of graphs and the field of real numbers. Any system of equations representing the vector space $\mathrm{WCW}(G)$ will be referred to as a *well-covering system* of G. (Precise definitions will be given in Section 9.1.)

In this work we study the problem called Well-Covering System: given a graph G, compute a well-covering system of G.

A graph is well-covered if and only if the vertex weight function that is constantly equal to 1 belongs to the well-covered vector space of the graph. Therefore, since the problem of recognizing well-covered graphs is co-NP-complete, the more general Well-Covering System problem is co-NP-hard.

The *well-covered dimension* of G is denoted by $\mathrm{wcdim}(G)$ and defined as the dimension of the well-covered vector space of G. Clearly, a graph G has well-covered dimension equal to zero if and only if the only well-covered weighting of G is the identically zero function. Such graphs are known to exist; for instance, the Petersen graph and any cycle with at least 8 vertices are among them (see [52, 54]). However, to the best of our knowledge, the complexity of computing the well-covered dimension of a graph is open, even in the special case of recognizing graphs with positive well-covered dimension. Caro and Yuster proved that the well-covered dimension of a tree is equal to the number of leaves [54]. Brown and Nowakowski generalized this result to the class of chordal graphs [50] by showing that in this case the well-covered dimension equals the number of simplicial cliques. They also showed that the well-covered dimension can be computed in polynomial time for cographs, for graphs with independence number at most two, and for chordal graphs. The well-covered dimension of certain product graphs was studied by Birnbaum et al. [26] and for Levi graphs of point-line configurations by Hauschild et al. [138].

Well-covered vector spaces of graphs containing no cycles of length 4 were studied by Brown, Nowakowski, and Zverovich [51]. Well-Covering System can be solved in polynomial time in classes of graphs of bounded vertex degree, as shown by Caro, Ellingham, and Ramey [52], in the class of graphs with girth at least 7, as shown by Caro and Yuster [54], and, as shown by Levit and Tankus, in the class of claw-free graphs [164] and in the class of graphs without cycles of lengths 4, 5, and 6 [165].

In the thesis we give two general reductions for the Well-Covering System problem, one based on modular decomposition and one based on anti-neighborhoods. Building on these results, we develop a polynomial-time algorithm for solving the problem in the class of fork-free graphs, thereby generalizing the analogous result of Levit and Tankus on claw-free graphs [164]. The algorithm decomposes a given fork-free graph G into a polynomial number of induced subgraphs of G and recursively computes a well-covering system for every graph H constructed at some step of the decomposition of G. To keep the well-covering systems polynomially bounded in size, Gaussian elimination is applied at each step. In the base case, when the subgraph H cannot be decomposed further, we use a structural result on fork-free graphs due to Lozin and Milanič [167, 168] (see also [97]) to infer that H is claw-free; hence, in this case the algorithm of Levit and Tankus applies.

The class of fork-free graphs generalizes also the class of cographs, hence our results generalize the result of Brown and Nowakowski [50] that the well-covered dimension of cographs can be computed in polynomial time. Furthermore, our reduction involving modular decomposition generalizes the analogous reduction for the (unweighted) well-covered graphs due to Klein, de Mello, and Morgana [147], who used modular and primeval decompositions to develop efficient algorithms for the problem of recognizing well-covered graphs in several extensions of the class of cographs.

8.2 Fair Allocation of Indivisible Items

Allocating resources to several agents in a satisfactory way is a classical problem in combinatorial optimization. Usually, such problems are equipped with some additional constraints for a feasible allocation, and there are various models of preferences expressed by the agents and different objectives arising from these. Here we study the fair allocation of n indivisible goods or items to a set of k agents. Each agent has its own additive utility function over the set of items. The goal is to assign every item to exactly one of the agents such that the minimal utility over all agents is as large as possible. In the area of Combinatorial Optimization a similar problem is well-known as the Santa Claus problem (see [15]), which can also be seen as a scheduling problem. In particular, Santa Claus asks for a distribution of items (presents) to agents (kids) so that the least happy kid is as happy as possible. Related problems of fair allocation are frequently studied in Computational Social Choice, see, e.g., [44]. Recent papers from this field containing many pointers to the literature and studying fairness issues, also in connection with an underlying graph structure, are given by [16, 43].

We look at the problem from a graph theoretical perspective and add a major new aspect to it. We allow an incompatibility relation between pairs of items, meaning that incompatible items should not be allocated to the same agent. This can reflect the fact that items rule out their joint usage or simply the fact that certain items are identical (or of a similar type) and it does not make sense for one agent to receive more than one of these items. We represent such a relation by a *conflict graph* where vertices correspond to items and edges express incompatibilities. If two items i and j are joined by an edge $ij \in E$, then i and j should not be allocated to a same agent.

In the literature we can find various combinatorial optimization problems equipped with conflict graphs leading to new problems with feasible solutions consisting of objects whose graph representations are independent sets in the conflict graph, e.g., knapsack problems [197, 198], bin packing [186], scheduling (see, e.g., [30, 101]), transportation [215] and problems on graphs (see, e.g., [89]).

Consider the example when items represent tasks with a starting and end time, and each agent should be allocated a fair subset of non-overlapping tasks. Then the mutual exclusion of two tasks will be represented by the edges of a conflict graph (see, e.g., [101, 173]). Another example is the transportation when orders need to be divided among several shipping partners which should all be treated as equally as possible, according to a joint agreement. In some industries, goods cannot be combined in an arbitrary way due to safety regulations, so a conflict graph can represent forbidden combinations of items. (see, e.g., [102, 142]).

Therefore, for a positive integer k we consider the problem called Fair k-Division Under Conflicts: given a graph G and k agents with utility functions over the set $V(G)$ of items, allocate items to agents so that the minimal value of utility over all agents is maximized. The formal definition of the problem will be given in Section 10.1. Clearly, every subset of items assigned to one agent has to form an independent set in th graph G, and in general, we allow the partial distribution of items, meaning that possibly not all items are distributed. The case when the conflict graph is edgeless corresponds to the problem Fair k-Division of Indivisible Items.

Note that for $k = 2$, the decision version of Fair k-Division of Indivisible Items (and thus of Fair k-Division Under Conflicts with edgeless conflict graph) also generalizes the decision version of the Knapsack problem: Given a set $V = \{1, \ldots, n\}$ of items with weights $w_1, \ldots, w_n \in \mathbb{Z}_+$ and values $v_1, \ldots, v_n \in \mathbb{Z}_+$, and two positive integers W and C such that $W < \sum_{j \in V} w_j$, is there a subset of the items having total weight at most W and total value at least C? It turns out that Fair k-Division of Indivisible Items, even with k identical profit functions, is weakly NP-hard for any constant $k \geq 2$ and strongly NP-hard for k being part of the input. Furthermore, we point out that Fair k-Division of Indivisible Items is still only weakly NP-hard for constant k even for arbitrary profit functions, since we can construct a pseudo-polynomial algorithm solving the problem with a k-dimensional dynamic programming array.

Note further that for $k = 1$, Fair k-Division Under Conflicts coincides with

Maximum Weight Independent Set. In particular, the case of unit weights and $k = 1$ generalizes Independent Set. Altogether, we conclude that Fair 1-Division Under Conflicts is strongly NP-hard. Hence, the addition of the conflict structure gives rise to a much more complicated problem, since Fair k-Division of Indivisible Items (which arises naturally as a special case for an edgeless conflict graph G) is trivial for $k = 1$ and only weakly NP-hard for $k \geq 2$.

In the thesis we aim to characterize the computational complexity of Fair k-Division Under Conflicts for different classes of conflict graphs. We study the boundary between strongly NP-hard cases and those where a pseudo-polynomial algorithm can be derived for a constant k. The fact that the Fair 2-Division Under Conflicts when a conflict graph is edgeless generalizes Knapsack problem implies that pseudo-polynomial algorithm is the only positive result we can obtain. Simultaneously, the proposed problem generalizes the Maximum Weight Independent Set problem, so pseudo-polynomial algorithms for the proposed problem can be obtained only in graph classes where the same holds for Maximum Weight Independent Set. We show that the problem is strongly NP-hard for bipartite graphs and their line graphs, and solvable in pseudo-polynomial time for the classes of chordal graphs, cocomparability graphs, biconvex bipartite graphs, graphs of bounded treewidth and graphs of bounded clique-width.

Chapter 9

Well-covered vector spaces in fork-free graphs

Our results from this chapter can be summarized as follows:

(1) We give two general reductions for the problem of computing the well-covered vector space of a given graph, one based on anti-neighborhoods and one based on modular decomposition, combined with Gaussian elimination.

(2) We develop a polynomial-time algorithm for computing the well-covered vector space of a given fork-free graph, generalizing the analogous result of Levit and Tankus for claw-free graphs.

(3) Our approach implies a polynomial-time recognition algorithm for the class of well-covered fork-free graphs and also generalizes some known results on cographs.

The results presented in this chapter are based on results from the following paper: Milanič, M., Pivač, N. Computing well-covered vector spaces of graphs using modular decomposition. Comp. Appl. Math. 42, 360 (2023). https://doi.org/10.1007/s40314-023-02502-8

9.1 Problem definition and preliminary remarks

Recall that given a weighted graph (G, w) and a set $S \subseteq V(G)$, the weight of S (with respect to w) is defined as $w(S) = \sum_{v \in S} w(v)$. Given a set $S \subseteq V(G)$, we denote by w_S the *restriction of* w *to* S, that is, the function $w_S : S \to \mathbb{R}$ defined by setting $w_S(v) = w(v)$ for all $v \in S$. Given a weighted graph (G, w), we say that w is a *well-covered weighting* of G and that G is w*-well-covered* if all maximal independent sets in G have the same weight with respect to w, that is, for every two maximal independent sets I and I' in G, we have $w(I) = w(I')$. Recall that for every graph G, the set $\mathrm{WCW}(G)$ of all well-covered weightings of G forms a vector space over the field of real numbers, called the *well-covered vector space* of G. Given a positive integer n, we denote by $[n]$ the set $\{1, \ldots, n\}$ (and $[0] := \emptyset$).

Since we only work with finite graphs, the well-covered vector space $\mathrm{WCW}(G)$ is always finite-dimensional and thus has a finite basis (an inclusion-wise maximal linearly independent set of vectors); furthermore, all bases of $\mathrm{WCW}(G)$ have the same cardinality, which is referred to as the *well-covered dimension* of G. Clearly, for every graph G, its well-covered dimension is an integer between 0 and $|V(G)|$.

Well-covered vector spaces of graphs can also be represented using systems of linear equations. Let G be a graph with n vertices. Fix an arbitrary ordering $v_1, \ldots, v_n$ of the vertices of G and an arbitrary ordering $I_1, \ldots, I_k$ of all maximal independent sets in G. By definition, a weight function $w : V(G) \to \mathbb{R}$ is a well-covered weighting of G if and only if w satisfies the following system of $\binom{k}{2}$ equations:

$$w(I_i) - w(I_j) = 0 \quad \text{for any two distinct } i, j \in [k] \text{ with } i < j\,. \tag{9.1}$$

To distinguish between vectors of abstract variables of a system and vectors of their concrete real values, we use the following convention throughout this chapter. To each vertex $v \in V(G)$ we associate a variable x_v, and write the systems of equations using such variable names. For example, following this convention, the system (9.1) corresponds to the following homogeneous linear system over the set of variables $\{x_v : v \in V(G)\}$:

$$\sum_{v \in I_i} x_v - \sum_{v \in I_j} x_v = 0 \quad \text{for any two distinct } i, j \in [k] \text{ with } i < j\,. \tag{9.2}$$

This system can be compactly represented with a single matrix equation

$$Ax = 0_r$$

where $r = \binom{k}{2}$, $A \in \mathbb{R}^{r \times n}$ is the coefficient matrix, and the right-hand side 0_r is the all-zero vector in $\mathbb{R}^r$. Thus, a column vector $w = (w(v_1), \ldots, w(v_n))^\top \in \mathbb{R}^n$ belongs to the well-covered vector space $\mathrm{WCW}(G)$ if and only if $Aw = 0_r$.

There are many ways to represent the well-covered vector space of a given graph G with a linear system. For example, a system equivalent to (9.2) with $k-1$ equations can be obtained by requiring that all maximal independent sets have the same weight as an arbitrary but fixed maximal independent set, say I_k:

$$\sum_{v \in I_i} x_v - \sum_{v \in I_k} x_v = 0 \quad \text{for all } i \in [k-1]\,. \tag{9.3}$$

Another equivalent system, also with $k-1$ equations, is the following:

$$\sum_{v \in I_i} x_v - \sum_{v \in I_{i+1}} x_v = 0 \quad \text{for all } i \in [k-1]\,. \tag{9.4}$$

A *well-covering system* of G is any system $\mathcal{S}$ of linear homogeneous equations over a set $\{x_v : v \in V(G)\}$ of variables indexed by the vertices of G such that a column vector $w = (w(v_1), \ldots, w(v_n))^\top \in \mathbb{R}^n$ belongs to the well-covered vector space $\mathrm{WCW}(G)$ if and only if it satisfies all the equations of the system. Given a well-covering system $\mathcal{S}$ of G, we denote by $|\mathcal{S}|$ the *size* of $\mathcal{S}$, that is, the number of equations in $\mathcal{S}$. As shown by systems (9.2) and (9.3), the same graph can admit well-covering systems of different sizes. Finally, we define the problem studied in this chapter.

Well-Covering System
Input: A graph $G = (V, E)$.
Task: Compute a well-covering system of G.

In the following sections we solve the WELL-COVERING SYSTEM in the class of fork-free graphs (see Fig. 9.1, part a)). Now we illustrate the main concepts with a concrete example and then discuss two important remarks about properties of well-covering systems.

Example 9.1.1. *Let G be the* bull graph*, that is, the graph obtained from the* 5*-vertex path with vertices $v_1, \dots, v_5$ in order along the path by adding to it the edge v_2v_4 (Fig. 9.1 b)). Then G has exactly three maximal independent sets: $I_1 = \{v_1, v_4\}$, $I_2 = \{v_2, v_5\}$, and $I_3 = \{v_1, v_3, v_5\}$. Any well-covered weighting w of G must satisfy that $w(I_1) = w(I_2) = w(I_3)$, or equivalently, $w(I_1) - w(I_3) = 0$ and $w(I_2) - w(I_3) = 0$. This yields the following linearly independent well-covering system $\mathcal{S}$ of G with size $r = 2$:*

$$\begin{aligned} - x_{v_3} + x_{v_4} - x_{v_5} &= 0 \\ -x_{v_1} + x_{v_2} - x_{v_3} \qquad\qquad &= 0\,. \end{aligned}$$

Using this system of equations we can easily determine for any weighting w of G whether it is well-covered weighting or not. For example, letting

$$\mathcal{B} = \{(1,1,0,0,0)^\top, (0,1,1,1,0)^\top, (0,0,0,1,1)^\top\}\,,$$

it can be easily verified that each $w = (w(v_1), \dots, w(v_5))^\top \in \mathcal{B}$ satisfies both equations in $\mathcal{S}$ and thus belongs to the space WCW(G)*. Furthermore, since the two rows of the coefficient matrix of the system $\mathcal{S}$, that is, $(0,0,-1,1,-1)$ and $(-1,1,-1,0,0)$, form a basis of the orthogonal complement of the well-covered vector space, it follows that the well-covered dimension of the space* WCW(G) *equals $|V(G)| - r = 3$. Thus, since the vectors in the set $\mathcal{B}$ are linearly independent, we infer that $\mathcal{B}$ is a basis of the well-covered vector space* WCW(G)*.* ▲

A remark on the size of well-covering systems. The number of maximal independent sets in an n-vertex graph can be exponential in n.[1] However, using Gaussian elimination it can be shown that any well-covering system of an n-vertex graph contains a well-covering subsystem of size at most n (see Lemma 9.1.2).

Consider an arbitrary well-covering system $\mathcal{S}$ of an n-vertex graph G and let r be the size of $\mathcal{S}$. Fix an arbitrary ordering of the vertices of G and an arbitrary ordering of the equations in $\mathcal{S}$. Let $A \in \mathbb{R}^{r \times n}$ be the coefficient matrix of $\mathcal{S}$. We say that a well-covering system $\mathcal{S}$ is *linearly independent* if the rows of the corresponding matrix A are linearly independent over the field of real numbers. In this case, the r rows of A form a basis of the orthogonal complement of the vector space WCW(G), and hence by standard linear algebra we have $r + \mathrm{wcdim}(G) = n$. In particular, in this case we have $r \le n$, and equality holds if and only if $\mathrm{wcdim}(G) = 0$, that is, the all-zero weighting is the only well-covered weighting of G.

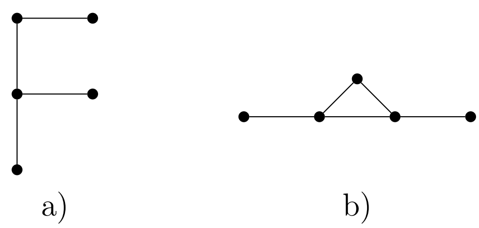


Figure 9.1: The fork and the bull.

[1]For example, the $2n$-vertex graph consisting of n isolated edges has 2^n maximal independent sets.

A remark on the coefficients of well-covering systems. Since we consider the well-covered vector space $\mathrm{WCW}(G)$ of a graph G as a vector space over the field of real numbers, any well-covering system of G consists of linear equations involving real numbers as coefficients. However, it often suffices to work with well-covering systems whose coefficients belong to a particular subset of the set of real numbers. We say that a well-covering system is *unit* if the matrix of the system has all the coefficients in the set $\{-1, 0, 1\}$, *integer* if the system consists of linear equations involving only integer coefficients, and *rational* if it consists of linear equations involving only rational coefficients. Note that systems (9.2), (9.3), and (9.4) as well as the system from Example 9.1.1 are all unit. Furthermore, the well-covering systems of fork-free graphs constructed by the algorithm given by our main result (Theorem 9.5.5, pg. 126) are also unit.

In some of our results, including the reduction based on modular decomposition (Theorem 9.2.7), the following lemma based on Gaussian elimination will be useful. We denote by $\omega < 2.3728596$ the matrix multiplication exponent (see, e.g., [7]).

Lemma 9.1.2. *Given an n-vertex graph G and a rational well-covering system $\widehat{\mathcal{S}}$ of G, one can compute in time $\mathcal{O}(n^{\omega-1}|\widehat{\mathcal{S}}|)$ a linearly independent well-covering system $\mathcal{S} \subseteq \widehat{\mathcal{S}}$ of G such that $|\mathcal{S}| \leq \min\{n, |\widehat{\mathcal{S}}|\}$.*

Proof. Let $r = |\widehat{\mathcal{S}}|$. If $r \leq n$, we are done, so assume $r > n$. Fix an arbitrary ordering of the vertices of G and an arbitrary ordering of the equations in $\widehat{\mathcal{S}}$. Let $A \in \mathbb{Q}^{r\times n}$ be the corresponding matrix and let $A^\top$ be its transpose. Using Gaussian elimination, we compute a basis B of $A^\top$ that is a maximal linearly independent subset of columns of $A^\top$. This can be done in time $\mathcal{O}(rn^{\omega-1})$ (see [58]). Note that the vectors in B correspond to certain equations in $\widehat{\mathcal{S}}$. Let $\mathcal{S} \subseteq \widehat{\mathcal{S}}$ consist of equations corresponding to the vectors in B. Then $\mathcal{S}$ is a linearly independent well-covering system of G, and clearly $|\mathcal{S}| \leq \min\{n, r\}$. Since $\omega \geq 2$ and the matrix A and its transpose can be computed in time $\mathcal{O}(rn)$, the algorithm runs in time $\mathcal{O}(rn^{\omega-1})$. □

9.2 Reduction to prime induced subgraphs

In this section we explain how to efficiently compute a well-covering system of a graph from well-covering systems of its maximal strong modules and of the representative graph. Then we combine this result with modular decomposition and Gaussian elimination to reduce the problem of computing a well-covering system of a graph to the same problem on certain prime induced subgraphs of the graph. For preliminaries on modular decomposition, see Section 2.4.

We start with a basic lemma characterizing the family of maximal independent sets in a graph G whose vertex set is equipped with an arbitrary partition into modules.

Lemma 9.2.1. *Let G be a graph, let $\mathcal{P} = \{M_1, \ldots, M_k\}$ be an arbitrary partition of $V(G)$ into modules, and let $G' = G/\mathcal{P}$ be the corresponding quotient graph, with $V(G') = \{v_1, \ldots, v_k\}$ where $v_j \in M_j$ for all $j \in [k]$. Then, a set $X \subseteq V(G)$ is a maximal independent set in G if and only if the following conditions hold.*

i) For all $j \in [k]$, the set $X \cap M_j$ is either empty or a maximal independent set in $G[M_j]$.

ii) The set $X' = \{v_j \in V(G') : X \cap M_j \neq \emptyset\}$ is a maximal independent set in G'.

Proof. First we show that the stated conditions are necessary. Let $X \subseteq V(G)$ be a maximal independent set in G. Consider an arbitrary $j \in [k]$ such that $X \cap M_j \neq \emptyset$. We want to prove that $X \cap M_j$ is a maximal independent set in $G[M_j]$. Since this set is a subset of X, it is an independent set in G and hence also in $G[M_j]$. We have to prove that it is a maximal one. Suppose for a contradiction that this is not the case, and let $x \in M_j \setminus X$ satisfy that $(X \cap M_j) \cup \{x\}$ is an independent set in $G[M_j]$. Then x has no neighbors in $X \cap M_j$. Since $X \cap M_j \neq \emptyset$, there exists a vertex $y \in S \cap M_j$. Note that x and y are in the same module M_j, so they have the same neighborhood outside M_j in G. In particular, this implies that $N_G(x) \cap (X \setminus M_j) = N_G(y) \cap (X \setminus M_j) \subseteq N_G(y) \cap X = \emptyset$, where the second equality follows from the fact that $y \in X$ and X is independent in G. We already know that x has no neighbors in $X \cap M_j$, so it follows that x has no neighbors in the set X at all. This implies that $X \cup \{x\}$ is the independent set in G, a contradiction with the maximality of X in G. Hence, condition *i*) holds.

Next we show condition *ii*), that is, the set $X' = \{v_j \in V(G') : X \cap M_j \neq \emptyset\}$ is a maximal independent set in G'. Let $J = \{j \in [k] : X \cap M_j \neq \emptyset\}$. Vertices in X are pairwise non-adjacent, so the modules M_j, $j \in J$, that contain vertices from X are anticomplete to each other in G. By construction of the graph G' it follows that the corresponding vertices $v_j, j \in J$, are pairwise non-adjacent in G', hence X' is an independent set in G'. It remains to prove maximality. Suppose for a contradiction that there is a vertex $v_\ell \in V(G') \setminus X'$ such that $X' \cup \{v_\ell\}$ is an independent set in G'. Since $v_\ell \notin S'$, it follows that $\ell \notin J$ and thus $X \cap M_\ell = \emptyset$. However, since $X' \cup \{v_\ell\}$ is an independent set in G', for all $j \in J$ we have that $v_\ell \notin N_{G'}(v_j)$, and it follows that modules M_ℓ and M_j are anticomplete to each other in G. Thus we can enlarge the independent set X in G by adding to it any vertex from M_j. This contradicts the fact that X is a maximal independent set in G.

The two conditions are also sufficient. Let $X \subseteq V(G)$ and assume that conditions *i*) and *ii*) from the lemma hold. We will prove that X is a maximal independent set in G. Let $J = \{j \in [k] : X \cap M_j \neq \emptyset\}$. Note that $X = \bigcup_{j \in J}(X \cap M_j)$. By condition *i*) we have that for all $j \in J$ the set $X \cap M_j \subseteq M_j$ is independent in G_j and hence in G. By condition *ii*) we have that the set $X' = \{v_j \in V(G') : j \in J\}$ is an independent set in G'. It follows that all the modules M_j, $j \in J$, are pairwise anticomplete. Hence the set $X = \bigcup_{j \in J}(X \cap M_j)$ is independent in G. It remains to show maximality. Suppose for a contradiction that there exists a vertex $v \in V(G) \setminus X$ such that the set $X \cup \{v\}$ is independent in G. Let $\ell \in [k]$ such that $v \in M_\ell$. Then $(X \cup \{v\}) \cap M_\ell$ is an independent set in G_ℓ, which implies that the set $X \cap M_\ell$ is not a maximal independent set in G_ℓ. By condition *i*) it follows that $X \cap M_\ell = \emptyset$ and thus $\ell \notin J$. Since the set $X \cup \{v\}$ is independent in G, the vertex v has no neighbors in the set $X = \bigcup_{j \in J}(X \cap M_j)$. As all the sets M_j are modules in G, this implies that v has no neighbors in the set $\bigcup_{j \in J} M_j$. Consequently, the vertex v_ℓ corresponding to the module M_ℓ in G' has no neighbors in the set $X' = \{v_j : j \in J\}$, in G'. Hence the set $X' \cup \{v_\ell\}$ is independent in G'. Since $v_\ell \notin X'$, this is a contradiction with the maximality of X', which is given by condition *ii*). This shows that the set X is a maximal independent set in G. □

We now use Lemma 9.2.1 to show how to efficiently compute a well-covering system of a graph from well-covering systems of its maximal strong modules and of the representative graph. We state the result more generally, for any graph equipped with a partition of the vertex set into modules, since we will later apply this result to various scenarios depending on whether the graph is disconnected (in which case the modules are the vertex sets of its connected components), the complement of the graph is disconnected (in which case the modules are the vertex sets of its cocomponents), or the graph and its complement are both connected.

Lemma 9.2.2. *Let G be a graph, let $\mathcal{P} = \{M_1, \ldots, M_k\}$ be an arbitrary partition of $V(G)$ into modules, and let $G' = G/\mathcal{P}$ be the corresponding quotient graph, with $V(G') = \{v_1, \ldots, v_k\}$ where $v_j \in M_j$ for all $j \in [k]$. Let $\mathcal{S}_j$ be a well-covering system for $G[M_j]$ for all $j \in [k]$ and let $\mathcal{S}'$ be a well-covering system of G'. Let $\mathcal{I} = \{I_j : j \in [k]\}$ be an arbitrary but fixed collection of maximal independent sets I_j in $G[M_j]$ for all $j \in [k]$. For each equation $s \in \mathcal{S}'$, let us denote by $\rho_{\mathcal{I}}(s)$ the equation indexed by the vertices of G obtained from s by iterating over all vertices v_j of G' and substituting the variable x_{v_j} corresponding to the vertex v_j with the sum $\sum_{v \in I_j} x_v$ (in particular, the variables corresponding to vertices v of G that do not belong to the union $\bigcup_{j \in [k]} I_j$ appear with zero coefficient). Then*

$$\mathcal{S} = \left(\bigcup_{j=1}^{k} \mathcal{S}_j\right) \cup \left\{\rho_{\mathcal{I}}(s) : s \in \mathcal{S}'\right\} \tag{9.5}$$

is a well-covering system of G. Furthermore, if the systems $\mathcal{S}_1, \ldots, \mathcal{S}_k$ and $\mathcal{S}'$ are all rational (resp. integer or unit), then so is $\mathcal{S}$.

Proof. Let G_j denote the graph $G[M_j]$ for all $j \in [k]$. The proof of the lemma will be based on the following observation.

Claim. *Let w be a vertex weight function on G, and let $w' : V(G') \to \mathbb{R}$ be defined as $w'(v_j) = w(I_j)$ for all $j \in [k]$. Let also w_j denote the restriction of w to $V(G_j)$ for all $j \in [k]$. Then G is w-well-covered if and only if G' is w'-well-covered and for all $j \in [k]$, the graph G_j is w_j-well-covered.*

Let us first show that the claim implies the lemma. We show that the proposed system of equations $\mathcal{S}$ given by (9.5) is a well-covering system of G by showing that, for any vertex weight function w on G, it holds that w is a well-covered weighting of G if and only if w satisfies all the equations of the system. Assume first that w is a well-covered weighting of G. Then, by the claim G' is w'-well-covered and for all $j \in [k]$, the graph G_j is w_j-well-covered. Since G' is w'-well-covered, w' is a solution of the system of equations $\mathcal{S}'$. Consider an arbitrary equation $s \in \mathcal{S}'$. Then there exist real numbers a_{v_j}, $j \in [k]$, such that s equals the equation $\sum_{j=1}^{k} a_{v_j} x_{v_j} = 0$. Hence, the equation $\rho_{\mathcal{I}}(s)$ is equivalent to the equation $\sum_{j=1}^{k} a_{v_j} \sum_{v \in I_j} x_v = 0$. Since setting $x_{v_j} = \sum_{v \in I_j} w(v)$ for all $v_j \in V(G')$ results in a solution of the equation s, we infer that setting $x_v = w(v)$ for all $v \in V(G)$ results in a solution of the equation $\rho_{\mathcal{I}}(s)$. Similarly, for each $j \in [k]$, setting $x_v = w_j(v) = w(v)$ for all $v \in V(G_j)$ yields a solution of the system of equations $\mathcal{S}_j$. It follows that setting $x_v = w(v)$ for all $v \in V(G)$ results in a solution of the system of equations $\bigcup_{j=1}^{k} \mathcal{S}_j$ and thus of the entire system of equations (9.5). Similar arguments show that if w is a solution of the system of equations (9.5), then w is a well-covered weighting of G. The last statement of the lemma, that the system $\mathcal{S}$ is rational (resp. integer or unit) whenever this is the case for the systems $\mathcal{S}_1, \ldots, \mathcal{S}_k$ and $\mathcal{S}'$, is straightforward.

Now we show the claim. Assume that G is w-well-covered. First we show that G' is w'-well-covered. Let I and I' be two maximal independent sets in G'. Let

$J = \{j \in [k] : v_j \in I\}$ and $J' = \{j \in [k] : v_j \in I'\}$ be the corresponding index sets. By Lemma 9.2.1, the sets $\bigcup_{j\in J} I_j$ and $\bigcup_{i\in J'} I_j$ are maximal independent sets in G. Since G is w-well-covered, it follows that $w\big(\bigcup_{j\in J} I_j\big) = w\big(\bigcup_{i\in J'} I_j\big)$. Furthermore, we have

$$w'(I) = \sum_{j\in J} w'(v_j) = \sum_{j\in J} w(I_j) = w\left(\bigcup_{j\in J} I_j\right)$$

and

$$w'(I') = \sum_{j\in J'} w'(v_j) = \sum_{j\in J'} w(I_j) = w\left(\bigcup_{j\in J'} I_j\right).$$

Altogether, the above equations imply that $w'(I) = w'(I')$ and since I and I' were arbitrary maximal independent sets in G', it follows that G' is w'-well-covered.

Next, we show that for all $j \in [k]$, the graph G_j is w_j-well-covered. Let I and I' be arbitrary maximal independent sets in G_j, and let S be a maximal independent set in G' such that $v_j \in S$. Let also $X = \bigcup\{I_\ell : v_\ell \in S \setminus \{v_j\}\}$. By Lemma 9.2.1, the sets $I \cup X$ and $I' \cup X$ are maximal independent sets in G. Since G is w-well-covered, we have that $w(I \cup X) = w(I' \cup X)$, and consequently

$$w_j(I) = w(I) = w(I \cup X) - w(X) = w(I' \cup X) - w(X) = w(I') = w_j(I').$$

Since I and I' were arbitrary maximal independent sets in G_j, we infer that G_j is w_j-well-covered.

For the proof of the other direction, assume that G' is w'-well-covered and that G_j is w_j-well-covered for all $j \in [k]$. We want to show that G is w-well-covered. Let I and I' be maximal independent sets in G, and let $J, J' \subseteq [k]$ be defined as $J = \{j \in [k] : I \cap M_j \neq \emptyset\}$ and $J' = \{j \in [k] : I' \cap M_j \neq \emptyset\}$. By Lemma 9.2.1, the sets $S = \{v_j \in V(G') : j \in J\}$ and $S' = \{v_j \in V(G') : j \in J'\}$ are maximal independent sets in G', and for all $j \in J$ (resp. $j \in J'$), the set $I \cap M_j$ (resp. $I' \cap M_j$) is a maximal independent set in G_j. Since for all $j \in [k]$ we have that G_j is w_j-well-covered, it follows that

$$w(I \cap M_j) = w_j(I \cap M_j) = w_j(I_j) = w(I_j) = w'(v_j) \text{ for all } j \in J$$

and similarly

$$w(I' \cap M_j) = w_j(I' \cap M_j) = w_j(I_j) = w(I_j) = w'(v_j) \text{ for all } j \in J'.$$

Thus, we have that $w(I) = \sum_{j\in J} w(I \cap M_j) = \sum_{j\in J} w'(v_j) = w'(S)$ and $w(I') = \sum_{j\in J'} w(I' \cap M_j) = \sum_{j\in J'} w'(v_j) = w'(S')$. Since G' is w'-well-covered, it follows that $w'(S) = w'(S')$ and consequently $w(I) = w(I')$, as we wanted to show. The sets I and I' were arbitrary maximal independent sets in G, hence it follows that G is w-well-covered. $\square$

We now apply Lemma 9.2.2 to three different cases: when G is disconnected, when the complement of G is disconnected, and when both G and its complement are connected.

Corollary 9.2.3. *Let G be a disconnected graph, with connected components $G_1, \ldots, G_k$ for some $k \geq 2$, and let $\mathcal{S}_j$ be a well-covering system of G_j for all $j \in [k]$. Then $\mathcal{S} = \bigcup_{j=1}^{k} \mathcal{S}_j$ is a well-covering system of G that can be computed in time $\mathcal{O}\big(\sum_{j=1}^{k} |\mathcal{S}_j|\big)$. Furthermore, if the systems $\mathcal{S}_1, \ldots, \mathcal{S}_k$ are all rational (resp. integer or unit), then so is $\mathcal{S}$.*

Proof. Let G be a graph with connected components $G_1,\ldots,G_k$. Then $\mathcal{P} = \{V(G_1),\ldots,V(G_k)\}$ is a partition of $V(G)$ into modules, and the corresponding quotient graph $G' = G/\mathcal{P}$ is the edgeless graph with k vertices. This implies that $V(G')$ is the only maximal independent set in G' and hence $\mathcal{S}' = \emptyset$ is a well-covering system of G'. By Lemma 9.2.2, it follows that the set $\bigcup_{j=1}^{k}\mathcal{S}_j$ is a well-covering system of G. This system can be computed in time $\mathcal{O}\big(\sum_{j=1}^{k}|\mathcal{S}_j|\big)$. □

Corollary 9.2.3 implies the fact that the well-covered dimension of a graph is the sum of the well-covered dimensions of its connected components (see [50]).

Corollary 9.2.4. *Let G be a graph with disconnected complement, with cocomponents $G_1,\ldots,G_k$, for some $k \geq 2$, and let $\mathcal{S}_j$ be a well-covering system of G_j for all $j \in [k]$. Let I_j be a maximal independent set in G_j for $j \in [k]$. Then*

$$\mathcal{S} = \left(\bigcup_{j=1}^{k}\mathcal{S}_j\right) \cup \left\{\sum_{v\in I_j} x_v - \sum_{v\in I_{j+1}} x_v = 0 : j \in [k-1]\right\}$$

is a well-covering system of G. In particular, given G, $G_1,\ldots,G_k$, and $\mathcal{S}_1,\ldots,\mathcal{S}_k$ as above, a well-covering system of G with size $\sum_{j=1}^{k}|\mathcal{S}_j| + k - 1$ can be computed in time $\mathcal{O}(|V(G)| + |E(G)| + \sum_{j=1}^{k}|\mathcal{S}_j|)$. Furthermore, if the systems $\mathcal{S}_1,\ldots,\mathcal{S}_k$ are all rational (resp. integer or unit), then so is $\mathcal{S}$.

Proof. Let G be a graph with cocomponents $G_1,\ldots,G_k$. Then $\mathcal{P} = \{V(G_1),\ldots,V(G_k)\}$ is a partition of $V(G)$ into modules, and the corresponding quotient graph $G' = G/\mathcal{P}$ is the complete graph on k vertices. Let $V(G') = \{v_1,\ldots,v_k\}$. Since G' is complete graph, the maximal independent sets in G' are exactly the singletons $\{v_j\}$ for $j \in [k]$. Consequently, w' is a well-covered weighting of G' if and only if $w'(v_1) = \ldots = w'(v_k)$, or equivalently, if for all $j \in [k-1]$ we have that $w'(v_j) = w'(v_{j+1})$. It follows that the set $\mathcal{S}' = \{x_{v_j} - x_{v_{j+1}} = 0 : j \in [k-1]\}$ is a well-covering system of G'. Let $\mathcal{I} = \{I_j : j \in [k]\}$. We follow the notation from Lemma 9.2.2 and for each $s \in \mathcal{S}'$ denote by $\rho_{\mathcal{I}}(s)$ the equation indexed by the vertices of G obtained from s by replacing each variable x_{v_j} corresponding to a vertex v_j of G' with the sum $\sum_{v\in I_j} x_v$. By Lemma 9.2.2 it follows that $\left(\bigcup_{j=1}^{k}\mathcal{S}_j\right) \cup \left\{\rho_{\mathcal{I}}(s) : s \in \mathcal{S}'\right\}$ is a well-covering system of G. Thus, the set $\left\{\rho_{\mathcal{I}}(s) : s \in \mathcal{S}'\right\}$ is equivalent to the set $\left\{\sum_{v\in I_j} x_v - \sum_{v\in I_{j+1}} x_v = 0 : j \in [k-1]\right\}$. It follows that $\mathcal{S} = \left(\bigcup_{j=1}^{k}\mathcal{S}_j\right) \cup \left\{\sum_{v\in I_j} x_v - \sum_{v\in I_{j+1}} x_v = 0 : j \in [k-1]\right\}$ is a well-covering system of G, as claimed. Furthermore, this system is integer, resp. unit, if the systems $\mathcal{S}_1,\ldots,\mathcal{S}_k$ are integer, resp. unit.

It remains to justify the time complexity. First, we compute for all $j \in [k]$ a maximal independent set I_j in the graph G_j. This can be done using a straightforward greedy algorithm in time $\mathcal{O}(\sum_{j=1}^{k}(|V(G_j)| + |E(G_j)|)) = \mathcal{O}(|V(G)| + |E(G)|)$. We compute the system of equations $\bigcup_{j=1}^{k}\mathcal{S}_j$ in time $\mathcal{O}(\sum_{j=1}^{k}|\mathcal{S}_j|)$ and the system of equations $\left\{\sum_{v\in I_j} x_v - \sum_{v\in I_{j+1}} x_v = 0 : j \in [k-1]\right\}$ in time $\mathcal{O}\Big(\sum_{j=1}^{k-1}(|I_{j+1}| + |I_j|)\Big) = \mathcal{O}\Big(\sum_{j=1}^{k}|I_j|\Big) = \mathcal{O}(|V(G)|)$. The total time complexity is $\mathcal{O}(|V(G)| + |E(G)| + \sum_{j=1}^{k}|\mathcal{S}_j|)$, as claimed. □

In the case when the graph and its complement are both connected, the corresponding algorithmic consequence of Lemma 9.2.2 is as follows.

Corollary 9.2.5. *Let $G = (V, E)$ be a connected and coconnected graph, let $\{M_1, \ldots, M_k\}$ be the partition of $V(G)$ into maximal strong modules, and let G' be the representative graph of G. Let I_j be a maximal independent set in the graph $G[M_j]$, let $\mathcal{S}_j$ be a well-covering system for $G[M_j]$ for all $j \in [k]$, and let $\mathcal{S}'$ be a well-covering system of G'. Then a well-covering system $\mathcal{S}$ of G with size $\sum_{j=1}^{k} |\mathcal{S}_j| + |\mathcal{S}'|$ can be computed in time $\mathcal{O}\big(|V| \cdot |\mathcal{S}'| + \sum_{j=1}^{k} |\mathcal{S}_j|\big)$. Furthermore, if the systems $\mathcal{S}_1, \ldots, \mathcal{S}_k$ and $\mathcal{S}'$ are all rational (resp. integer or unit), then so is $\mathcal{S}$.*

Proof. Let $\mathcal{I} = \{I_j : j \in [k]\}$. Using the notation of Lemma 9.2.2, the lemma implies that it suffices to compute the system of equations $\mathcal{S} = \big(\bigcup_{j=1}^{k} \mathcal{S}_j\big) \cup \big\{\rho_{\mathcal{I}}(s) : s \in \mathcal{S}'\big\}$. This can be done in time

$$\mathcal{O}\left(\sum_{j=1}^{k} |\mathcal{S}_j| + |\mathcal{S}'| \left(\sum_{j=1}^{k} |I_j|\right)\right) = \mathcal{O}\left(\sum_{j=1}^{k} |\mathcal{S}_j| + |V| \cdot |\mathcal{S}'|\right),$$

as claimed. □

For the proof of main result of this section, recall that a leaf of a rooted tree T is a node without any successors, while an internal node of T is a node that is not a leaf. Note that if T is a one-vertex rooted tree, then the unique vertex in T is both the root and a leaf of T, but it is not an internal node. Given a rooted tree T, we denote by $\ell(T)$ the number of leaves of T and by $i(T) = |V(T)| - \ell(T)$ the number of internal nodes of T. We will need the following well-known property of rooted trees. To keep the thesis self-contained, we include a proof.

Lemma 9.2.6. *Let T be a rooted tree in which each internal node has at least two successors. Then $\ell(T) \geq i(T) + 1$.*

Proof. By induction on $i(T)$. If $i(T) = 0$, then the unique vertex in T is a leaf and the inequality holds. Let now T be a tree with $i(T) \geq 1$ such that each internal node of T has at least two successors, and assume that every tree T' with $i(T') < i(T)$ in which each internal node has at least two successors satisfies that $\ell(T') \geq i(T') + 1$. Let r be the root of T and let d be the number of successors of r. Since $i(T) \geq 1$, the root of T is an internal node. Hence $d \geq 2$. Let $T_1, \ldots, T_d$ be the rooted trees obtained by the deletion of r from T, where for each $j \in [r]$, the root of T_j is the unique successor of r in T_j. Then for all $j \in [r]$ we have that $i(T_j) < i(T)$, and by the induction hypothesis every T_j satisfies that $\ell(T_j) \geq i(T_j) + 1$. Observe that every internal node of T_j, $j \in [d]$, is also internal in T, and node r is internal in T as well, so we have $i(T) = 1 + \sum_{j=1}^{d} i(T_j)$. Since $\ell(T) = \sum_{j=1}^{d} \ell(T_j)$, we conclude that $\ell(T) \geq \sum_{j=1}^{d} (i(T_j) + 1) = \sum_{j=1}^{d} i(T_j) + d \geq (i(T) - 1) + 2 = i(T) + 1$, which completes the proof. □

We now prove the main result of this section, a reduction of the problem of computing a well-covering system of a graph to the same problem on certain prime induced subgraphs of the graph. We say that a function $f : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is *nondecreasing* if $0 \leq x_1 \leq x_2$ and $0 \leq y_1 \leq y_2$ implies $f(x_1, y_1) \leq f(x_2, y_2)$, and *superadditive* if the inequality

$$f(x_1, y_1) + f(x_2, y_2) \leq f(x_1 + x_2, y_1 + y_2)$$

holds for all $x_1, y_1, x_2, y_2 \in \mathbb{R}^+$. Note that every superadditive function is nondecreasing.

Theorem 9.2.7. *Let $\mathcal{G}$ be a class of graphs and $\mathcal{G}^*$ the class of all prime induced subgraphs of graphs in $\mathcal{G}$. Assume that for each graph G in $\mathcal{G}^*$ with n vertices and $m \geq 1$ edges one can compute in time $f(n,m)$ a rational (resp. integer or unit) well-covering system of G with size at most n, where f is a superadditive function. Then for any graph G in $\mathcal{G}$ with n vertices and m edges, one can compute in time $\mathcal{O}\big(f(2n,m)+n^{\omega+1}\big)$ a rational (resp. integer or unit) well-covering system of G with size at most n.*

Proof. Let G be a graph in $\mathcal{G}$ with n vertices and m edges. Let T_G be the modular decomposition tree of G. This tree can be computed in time $\mathcal{O}(n+m)$ [177]. Recall that for a node t of T_G, we denote by G_t the subgraph of G induced by the vertices appearing in the one-vertex subgraphs labeling the leaves of the subtree of T_G rooted at t. Let $n_t = |V(G_t)|$ and $m_t = |E(G_t)|$.

We traverse the tree T_G bottom-up and for each node $t \in V(T_G)$ we recursively compute a maximal independent set I_t in G_t and a well-covering system $\mathcal{S}_t$ of G_t with size at most n_t. It is important to note that we do not store a complete representation of the graph G_t via adjacency lists, as that would additionally increase the time and space complexity of the procedure. The ordering in which the nodes of tree T_G are traversed can be computed in time $\mathcal{O}(|V(T_G)|) = \mathcal{O}(n+m)$, for example, by reversing the ordering in which the nodes of T_G are visited by a breadth-first search from the root node. For each node t of T_G, we denote by C_t the set of all children of t in T_G.

Assume first that t is a leaf node (that is, $C_t = \emptyset$). Then $V(G_t) = \{v_t\}$ where v_t is the vertex of G labeling t; in particular, $n_t = 1$. Hence, $I_t = V(G_t)$ is the only maximal independent set in G_t and $\mathcal{S}_t = \emptyset$ is a well-covering system of G_t that trivially satisfies the inequality $|\mathcal{S}_t| \leq n_t$. Both I_t and $\mathcal{S}_t$ can be computed in constant time.

Assume now that t is an internal node in T_G. Then t is one of the types parallel, series, or prime. Since the subtrees of T_G rooted at the children of t are the modular decomposition trees of the subgraphs of G_t induced by its maximal strong modules, which form a partition of $V(G_t)$, it follows that $n_t = \sum_{u \in C_t} n_u$. For each child u of t we have already computed a maximal independent set I_u in G_u and a well-covering system $\mathcal{S}_u$ of G_u with size at most n_u. We explain how to efficiently combine these into a maximal independent set I_t in G_t and a well-covering system $\mathcal{S}_t$ of G_t with size at most n_t for each of the three cases separately.

- If t is of type parallel, then G_t is a disconnected graph, with connected components G_u, $u \in C_t$. We can thus take $I_t = \bigcup_{u \in C_t} I_u$ and by Corollary 9.2.3, $\mathcal{S}_t = \bigcup_{u \in C_t} \mathcal{S}_u$. We have

$$|\mathcal{S}_t| = \sum_{u \in C_t} |\mathcal{S}_u| \leq \sum_{u \in C_t} n_u = n_t.$$

  Furthermore, by Corollary 9.2.3 the well-covering system $\mathcal{S}_t$ of G_t can be computed in time $\mathcal{O}\big(\sum_{u \in C_t} |\mathcal{S}_u|\big) = \mathcal{O}\big(|\mathcal{S}_t|\big) = \mathcal{O}(n_t)$. Since I_t can be computed in time $\mathcal{O}(|V(G_t)|+|E(G_t)|) = \mathcal{O}(n_t+m_t)$, the total time complexity at the parallel node t is $\mathcal{O}(n_t+m_t)$.

- If t is of type series, then the complement of G_t is disconnected, with cocomponents G_u, $u \in C_t$. We select an arbitrary $u \in C_t$ and set $I_t = I_u$. Furthermore, we fix an arbitrary ordering $u_1, \ldots, u_p$ of the set C_t and set

$$\widehat{\mathcal{S}_t} = \left(\bigcup_{u \in C_t} \mathcal{S}_u\right) \cup \left\{ \sum_{v \in I_{u_j}} x_v - \sum_{v \in I_{u_{j+1}}} x_v = 0 : j \in [p-1] \right\}.$$

By Corollary 9.2.4, $\widehat{\mathcal{S}_t}$ is a well-covering system of G_t that can be computed in time $\mathcal{O}(|V(G_t)| + |E(G_t)| + \sum_{u \in C_t} |\mathcal{S}_u|) = \mathcal{O}(n_t + m_t)$. The size of $\widehat{\mathcal{S}_t}$ is bounded as follows:

$$|\widehat{\mathcal{S}_t}| = \sum_{u \in C_t} |\mathcal{S}_u| + |C_t| - 1 \leq \sum_{u \in C_t} n_u + n_t - 1 = n_t + n_t - 1 = 2n_t - 1.$$

Furthermore, Lemma 9.1.2 implies that a well-covering system $\mathcal{S}_t \subseteq \widehat{\mathcal{S}_t}$ of G_t such that $|\mathcal{S}_t| \leq n_t$ can be computed in time $\mathcal{O}(n_t^{\omega-1} \cdot |\widehat{\mathcal{S}_t}|) = \mathcal{O}(n_t^{\omega})$. Altogether, this implies that the independent set I_t and a well-covering system $\mathcal{S}_t$ of G_t with size at most n_t at the series node t can be computed in time $\mathcal{O}(n_t + m_t + n_t^{\omega}) = \mathcal{O}(n_t^{\omega})$ (since $\omega \geq 2$).

- Consider now the case when the node t is of type prime. In this case, the graph H_t labeling the node t is a prime induced subgraph of G_t and hence of G. Each child u of t in T_G corresponds to a unique maximal strong module M_u of G. The graph H_t is the representative graph of G_t, hence it contains a unique vertex v_u from each maximal strong module M_u of G_t.

  Since H_t is a prime induced subgraph of G, it belongs to $\mathcal{G}^*$ and hence, a well-covering system $\mathcal{S}'$ of H_t with size at most $|V(H_t)|$ can be computed in time $f(|V(H_t)|, |E(H_t)|)$. Next, we compute in time $\mathcal{O}(|V(H_t)| + |E(H_t)|)$ a maximal independent set I'_t in H_t. Let $C'_t = \{u \in C_t : v_u \in I'_t\}$. By Lemma 9.2.1, the set $I_t = \bigcup_{u \in C'_t} I_u$ is a maximal independent set in G_t. By Corollary 9.2.5, a well-covering system $\widehat{\mathcal{S}_t}$ of G_t with size $\sum_{u \in C_t} |\mathcal{S}_u| + |\mathcal{S}'|$ can be computed in time $\mathcal{O}\big(|V(G_t)| \cdot |\mathcal{S}'| + \sum_{u \in C_t} |\mathcal{S}_u|\big)$. Since $|\mathcal{S}'| \leq |V(H_t)| \leq n_t$, it follows that $|\widehat{\mathcal{S}_t}| \leq \sum_{u \in C_t} n_u + n_t = n_t + n_t = 2n_t$.

  Using Lemma 9.1.2, a well-covering system $\mathcal{S}_t \subseteq \widehat{\mathcal{S}_t}$ of G_t such that $|\mathcal{S}_t| \leq n_t$ can be computed in time $\mathcal{O}\big(n_t^{\omega-1}|\widehat{\mathcal{S}_t}|\big) = \mathcal{O}(n_t^{\omega})$. The total time complexity of computing $\mathcal{S}_t$ at the node t is

$$\begin{aligned}
&\mathcal{O}\left(f(|V(H_t)|, |E(H_t)|) + |V(G_t)| \cdot |\mathcal{S}'| + \sum_{u \in C_t} |\mathcal{S}_u| + n_t^{\omega} \right) \\
=\ &\mathcal{O}(f(|V(H_t)|, |E(H_t)|) + n_t^2 + n_t + n_t^{\omega}) \\
=\ &\mathcal{O}(f(|V(H_t)|, |E(H_t)|) + n_t^{\omega}),
\end{aligned}$$

  while the independent set I_t can be computed in time $\mathcal{O}(|V(H_t)| + |E(H_t)| + |V(G_t)|) = \mathcal{O}(n_t + m_t)$.

  Thus, the total time complexity at the prime node t is $\mathcal{O}(f(|V(H_t)|, |E(H_t)|) + n_t^{\omega})$.

It remains to sum up the time complexities over all nodes of T_G. We compute separately the sum over all leaves of T_G and over all internal nodes of T_G. Let us denote by L the set of all leaves of T_G. Recall that by the definition of a modular decomposition tree, the leaves of T_G are in a bijective correspondence with the vertices of G, and thus $|L| = n$. By Lemma 9.2.6 it follows that the number of internal nodes of T_G is at most $n-1$. Note also that for each internal node t, the number of vertices of H_t equals the number of children of t in T_G, which implies that the total number of vertices of the graphs H_t, summed up over all internal nodes t, equals the number of edges of T_G, which is at most $|L| + |V(T_G) \setminus L| - 1 \leq n + (n-1) - 1 = 2n-2$. Furthermore, for each internal node t, the edges of H_t correspond to distinct edges of G (joining two vertices of G_t from distinct maximal strong modules), and no two edges from representative graphs of two different internal nodes correspond to the same edge of G. This implies that the total number of edges of the graphs H_t, summed up over all internal nodes t, is at most m.

We already saw that in each leaf t of T_G the algorithm computes the independent set I_t and the well-covering system $\mathcal{S}_t$ in constant time. Hence, summing over all leaves of T_G we obtain the time complexity of $\mathcal{O}(n)$. If t is an internal node, then the algorithm computes I_t and $\mathcal{S}_t$ in time $\mathcal{O}(n_t + m_t)$ if t is of type parallel, in time $\mathcal{O}(n_t^\omega)$ if t is of type series, and in time $\mathcal{O}(f(|V(H_t)|, |E(H_t)|) + n_t^\omega)$ if t is of type prime. Furthermore, $|E(H_t)| \leq m$.

The sum of time complexities over all the internal nodes of T_G can thus be bounded as follows.

$$\begin{aligned}
&\mathcal{O}\left(\sum_{t\in V(T_G)\setminus L} \left(f(|V(H_t)|, |E(H_t)|) + n_t^\omega\right)\right)\\
=&\,\mathcal{O}\left(f\left(\sum_{t\in V(T_G)\setminus L} |V(H_t)|, \sum_{t\in V(T_G)\setminus L} |E(H_t)|\right) + \sum_{t\in V(T_G)\setminus L} n_t^\omega\right) =\\
=&\,\mathcal{O}\left(f(2n, m) + n^{\omega+1}\right),
\end{aligned}$$

where the first equality holds since f is a superadditive function and the last one since $\sum_{t\in V(T_G)\setminus L} |V(H_t)| \leq 2n-2$, $\sum_{t\in V(T_G)\setminus L} |E(H_t)| \leq m$, and f is nondecreasing. Since the time complexity over all leaves of T_G is $\mathcal{O}(n)$, the total time complexity over all nodes in T_G is equal to $\mathcal{O}\big(f(2n,m) + n^{\omega+1}\big)$. Finally, recall that the algorithm first needs $\mathcal{O}(n+m)$ time to compute the modular decomposition tree T_G and an ordering in which the nodes of T_G are visited. Thus, altogether, the algorithm runs in time $\mathcal{O}\big(n + m + f(2n,m) + n^{\omega+1}\big) = \mathcal{O}\big(f(2n,m) + n^{\omega+1}\big)$.

□

Remark 9.2.8. One of the assumptions in Theorem 9.2.7 is that for each graph G in $\mathcal{G}^*$ with n vertices and m edges one can compute in time $f(n,m)$ a well-covering system of G with size at most n. If instead, only an algorithm is available for computing an arbitrary rational (resp. integer or unit) well-covering system of $G \in \mathcal{G}^*$ in time $f(n,m)$ (that is, without a bound of n on the size of the system), then one can first combine such an algorithm with Lemma 9.1.2. This would result in an algorithm that, given a graph G from $\mathcal{G}$ with n vertices and m edges, in time $\mathcal{O}\big(f(2n,m)\cdot n^{\omega-1} + n^{\omega+1}\big)$ computes a rational (resp. integer or unit) well-covering system of G with size at most n.

9.3 Cographs

The proof of Theorem 9.2.7 relies on Gaussian elimination. If the input graph possesses some additional combinatorial structure, the use of Gaussian elimination may be avoided, and this can lead to faster algorithms. As we show in this section, this is the case for the class of *cographs*. Cographs are defined as graphs that can be constructed starting from copies of the one-vertex graph using the operations of disjoint union and complementation (see, e.g., [47]). Thus, the only prime cograph is the one-vertex graph, and the modular decomposition tree of a cograph contains only parallel and series nodes.

It follows from Theorem 9.2.7 that a well-covering system of a given cograph G with n vertices can be computed in time $\mathcal{O}(n^{\omega+1})$. We improve this time complexity as follows.

Theorem 9.3.1. *Given a cograph G with n vertices and m edges, an integer well-covering system of G with size at most $n-1$ can be computed in time $\mathcal{O}(n(n+m))$.*

Proof. Let G be a cograph with n vertices and m edges. Let T_G be the modular decomposition tree of G. As before, given a node $t \in V(T_G)$, we denote by G_t the subgraph of G induced by the vertices appearing in the one-vertex subgraphs labeling the leaves of the subtree of T_G rooted at t. Let $n_t = |V(G_t)|$ and $m_t = |E(G_t)|$. Since G is a cograph, every internal node of T_G is of type either parallel or series. We traverse the tree T_G bottom-up and for each node $t \in V(T_G)$ we recursively compute a maximal independent set I_t in G_t and a well-covering system $\mathcal{S}_t$ of G_t with size at most $n_t - 1$. For each node t of T_G, we denote by C_t the set of all children of t in T_G.

If t is a leaf node (that is, $C_t = \emptyset$), then $I_t = V(G_t)$ is a maximal independent set of G_t and $\mathcal{S}_t = \emptyset$ is a well-covering system of G_t, with size $0 = n_t - 1$. Both I_t and $\mathcal{S}_t$ can be computed in constant time. If t is an internal node in T_G, then t is of type either parallel or series. For each child u of t we have already computed a maximal independent set I_u in G_u and a well-covering system $\mathcal{S}_u$ of G_u with size at most $n_u - 1$. We explain how to efficiently combine these into a maximal independent set I_t in G_t and a well-covering system $\mathcal{S}_t$ of G_t with size at most $n_t - 1$ for both cases. If t is of type parallel, then G_t is a disconnected graph, with connected components G_u, $u \in C_t$. We can thus take $I_t = \bigcup_{u\in C_t} I_u$ and by Corollary 9.2.3, $\mathcal{S}_t = \bigcup_{u\in C_t} \mathcal{S}_u$. We have

$$|\mathcal{S}_t| = \sum_{u\in C_t} |\mathcal{S}_u| \le \sum_{u\in C_t} (n_u - 1) = n_t - |C_t| \le n_t - 1.$$

Furthermore, by Corollary 9.2.3 the well-covering system $\mathcal{S}_t$ of G_t can be computed in time $\mathcal{O}\big(\sum_{u\in C_t} |\mathcal{S}_u|\big) = \mathcal{O}\big(|\mathcal{S}_t|\big) = \mathcal{O}(n_t)$. Since I_t can be computed in time $\mathcal{O}(|V(G_t)| + |E(G_t)|) = \mathcal{O}(n_t + m_t)$, the total time complexity at the parallel node t is $\mathcal{O}(n_t + m_t)$. If t is of type series, then the complement of G_t is disconnected, with cocomponents G_u, $u \in C_t$. We fix an arbitrary ordering $u_1, \ldots, u_p$ of the set C_t of children of t and obtain the new maximal independent set I_t and a well-covering system $\mathcal{S}_t$ by setting $I_t = I_{u_1}$ and

$$\mathcal{S}_t = \left(\bigcup_{u\in C_t} \mathcal{S}_u\right) \cup \left\{ \sum_{v\in I_{u_j}} x_v - \sum_{v\in I_{u_{j+1}}} x_v = 0 : j \in [p-1] \right\}.$$

By Corollary 9.2.4, the system $\mathcal{S}_t$ is indeed a well-covering system of G_t and can be computed in time $\mathcal{O}(|V(G_t)| + |E(G_t)| + \sum_{u\in C_t} |\mathcal{S}_u|) = \mathcal{O}(n_t + m_t)$. The size of $\mathcal{S}_t$ is bounded as follows:

$$|\mathcal{S}_t| = \sum_{u\in C_t} |\mathcal{S}_u| + |C_t| - 1 \leq \sum_{u\in C_t} (n_u - 1) + |C_t| - 1 = n_t - 1.$$

Altogether, this implies that the independent set I_t and a well-covering system $\mathcal{S}_t$ of G_t with size at most $n_t - 1$ at the series node t can be computed in time $\mathcal{O}(n_t + m_t)$.

Note that all the well-covering systems computed by the algorithm are integer. It remains to estimate the time complexity of the algorithm. The tree T_G can be computed in time $\mathcal{O}(n+m)$ [177], and in the same time we can compute an ordering in which the nodes of tree T_G are traversed. Recall that the number of leaves of T_G is equal to n, while from Lemma 9.2.6 it follows that the number of internal nodes of T_G is at most $n - 1$. We already saw that in each leaf t of T_G the algorithm spends only constant time, while in each internal node t of T_G the independent set I_t and a well-covering system $\mathcal{S}_t$ of G_t can be computed in time $\mathcal{O}(n_t + m_t)$. Summing over all nodes of T_G we get the time complexity $\mathcal{O}(n + (n-1)\cdot(n+m)) = \mathcal{O}(n(n+m))$. We infer that the total time complexity of the algorithm is $\mathcal{O}(n(n+m))$. □

Let us mention two consequences of Theorem 9.3.1.

First, applying the theorem to a given n-vertex cograph G, we obtain in polynomial time an integer well-covering system $\widehat{\mathcal{S}}$ with size at most $n - 1$. Using Gaussian elimination (cf. Lemma 9.1.2), we can then compute in time $\mathcal{O}(n^{\omega})$ a linearly independent well-covering subsystem $\mathcal{S} \subseteq \widehat{\mathcal{S}}$ of G. Consequently, we can compute the well-covered dimension of G as the difference $n - |\mathcal{S}|$. This implies a result of Brown and Nowakowski [50] who showed that the well-covered dimension of cographs can be computed in polynomial time.

Second, a graph G has well-covered dimension equal to zero if and only if the only well-covered weighting of G is the identically zero function, or, equivalently, if G admits no well-covering system with size less than n. Therefore, Theorem 9.3.1 implies the following.

Corollary 9.3.2. *Every cograph has a strictly positive well-covered dimension.*

An alternative proof of this result could be obtained by using the fact that every cograph is equistable (see [171]).

9.4 Reduction to anti-neighborhoods

In this section we focus on the subgraphs of a given graph obtained by the deletion of the closed neighborhood of some vertex in the graph. Given a graph G with vertex set $\{v_1, \ldots, v_n\}$, we denote by G_j the graph $G - N[v_j]$, for all $j \in [n]$. We first show that, given a well-covering system of the graph G_j, for all $j \in [n]$, we can efficiently compute a well-covering system of G.

Lemma 9.4.1. *Let G be a graph with vertex set $\{v_1, \ldots, v_n\}$. For each $j \in [n]$ let $\mathcal{S}_j$ be a rational (resp. integer or unit) well-covering system of $G - N[v_j]$ and I_j a maximal* *independent set of $G - N[v_j]$. Then*

$$\left(\bigcup_{j=1}^{n} \mathcal{S}_j\right) \cup \left\{ \sum_{v \in I_j \cup \{v_j\}} x_v - \sum_{v \in I_{j+1} \cup \{v_{j+1}\}} x_v = 0 : j \in [n-1] \right\}$$

is a rational (resp. integer or unit) well-covering system of G.

Proof. Let G be a graph and let w be a vertex weight function on G. For each $j \in [n]$ let G_j denote the graph $G - N[v_j]$ and w_j the restriction of w to $V(G_j)$. We show the following claim: G is w-well-covered if and only if for all $j \in [n]$ it holds that G_j is w_j-well-covered and for all $j \in [n-1]$ it holds that $w(I_j \cup \{v_j\}) = w(I_{j+1} \cup \{v_{j+1}\})$. From the claim we get that the equations from the well-covering systems $\mathcal{S}_j$ of G_j, over all $j \in [n]$, along with the equations of the form

$$\sum_{v \in I_j \cup \{v_j\}} x_v - \sum_{v \in I_{j+1} \cup \{v_{j+1}\}} x_v = 0$$

for $j \in [n-1]$, form a well-covering system of G.

Let us prove the claim. Assume first that G is w-well-covered. Let $j \in [n]$ and let I and I' be maximal independent sets in G_j. Then the sets $I \cup \{v_j\}$ and $I' \cup \{v_j\}$ are maximal independent sets in G. Since G is w-well-covered, it holds that $w(I \cup \{v_j\}) = w(I' \cup \{v_j\})$. Consequently, we have that

$$w_j(I) = w(I) = w(I \cup \{v_j\}) - w(v_j) = w(I' \cup \{v_j\}) - w(v_j) = w(I') = w_j(I')\,,$$

and G_j is w_j-well-covered. Consider now an arbitrary $j \in [n-1]$. Since I_j and I_{j+1} are maximal independent sets in G_j and G_{j+1}, respectively, the sets $I_j \cup \{v_j\}$ and $I_{j+1} \cup \{v_{j+1}\}$ are maximal independent sets in G. Since G is w-well-covered, it follows that $w(I_j \cup \{v_j\}) = w(I_{j+1} \cup \{v_{j+1}\})$, which is what we wanted to show.

For a proof of the other direction, assume that for all $j \in [n]$ it holds that G_j is w_j-well-covered and for all $j \in [n-1]$ it holds that $w(I_j \cup \{v_j\}) = w(I_{j+1} \cup \{v_{j+1}\})$. In particular, this implies that $w(I_j \cup \{v_j\}) = w(I_k \cup \{v_k\})$ for all $j, k \in [n]$. We want to prove that G is w-well-covered. Let I and I' be maximal independent sets in G and let $v_j \in I$ and $v_k \in I'$. Note that $I \setminus \{v_j\}$ and $I' \setminus \{v_k\}$ are maximal independent sets in G_j and G_k, respectively. By assumption G_j is w_j-well-covered and G_k is w_k-well-covered, and thus we have that $w(I \setminus \{v_j\}) = w_j(I \setminus \{v_j\}) = w_j(I_j) = w(I_j)$ and, similarly, $w(I' \setminus \{v_k\}) = w(I_k)$. Consequently,

$$w(I) = w(I \setminus \{v_j\}) + w(v_j) = w(I_j) + w(v_j) = w(I_j \cup \{v_j\})$$

and

$$w(I') = w(I' \setminus \{v_k\}) + w(v_k) = w(I_k) + w(v_k) = w(I_k \cup \{v_k\})\,.$$

The above two expressions are equal by assumption, so we get $w(I) = w(I')$ and thus G is w-well-covered. □

Corollary 9.4.2. *Let G be a graph with vertex set $\{v_1, \ldots, v_n\}$. For each $j \in [n]$ let $\mathcal{S}_j$ be a rational (resp. integer or unit) well-covering system of $G - N[v_j]$. Then a rational (resp. integer or unit) well-covering system of G with size $\sum_{j=1}^{n} |\mathcal{S}_j| + n - 1$ can be computed in time $\mathcal{O}(n(n+m) + \sum_{j=1}^{n} |\mathcal{S}_j|)$, where $m = |E(G)|$.*

Proof. In time $\mathcal{O}(n(n+m))$ we compute the graphs $G - N[v_j]$ for all $j \in [n]$ and a maximal independent set I_j in each such graph. Then, using Lemma 9.4.1 we compute a well-covering system of G in time $\mathcal{O}(\sum_{j=1}^{n} |\mathcal{S}_j| + n^2)$. The total complexity of this approach is $\mathcal{O}(n(n+m) + \sum_{j=1}^{n} |\mathcal{S}_j|)$, as claimed. $\square$

Using the above result, we give a more general statement, which will be an ingredient of the main algorithm in this paper.

Theorem 9.4.3. *Let $\mathcal{G}$ and $\mathcal{G}^*$ be two graph classes such that for every graph G in $\mathcal{G}$ and every vertex v of G the graph $G - N[v]$ is in $\mathcal{G}^*$. Assume that for each graph G in $\mathcal{G}^*$ with n vertices and m edges one can compute in time $f(n,m)$ a rational (resp. integer or unit) well-covering system of G with size at most $g(n,m)$, where f and g are nondecreasing functions. Then for any graph G in $\mathcal{G}$ with n vertices and m edges, one can compute in time $\mathcal{O}(n \cdot (n+m+f(n,m)))$ a rational (resp. integer or unit) well-covering system of G with size at most $n \cdot g(n,m) + n - 1$.*

Proof. Let G be a graph in $\mathcal{G}$ with vertex set $V(G) = \{v_1, \ldots, v_n\}$ and let $m = |E(G)|$. For all $j \in [n]$, let $G_j = G - N[v_j]$. The graphs G_j, $j \in [n]$, can be computed in time $\mathcal{O}(n(n+m))$. By assumption, for each $j \in [n]$ the graph G_j is in $\mathcal{G}^*$, and hence a rational (resp. integer or unit) well-covering system $\mathcal{S}_j$ of G_j with at most $g(|V(G_j)|, |E(G_j)|) \leq g(n,m)$ equations can be computed in time $f(|V(G_j)|, |E(G_j)|) \leq f(n,m)$. Note also that $|\mathcal{S}_j| \leq f(|V(G_j)|, |E(G_j)|) \leq f(n,m)$. By Corollary 9.4.2, a well-covering system of G with size $\sum_{j=1}^{n} |\mathcal{S}_j| + n - 1 \leq n \cdot g(n,m) + n - 1$ can be computed in time $\mathcal{O}(n(n+m) + \sum_{j=1}^{n} |\mathcal{S}_j|) = \mathcal{O}(n \cdot (n+m+f(n,m)))$. $\square$

9.5 Fork-free graphs

By Theorem 9.3.1, a well-covering system of a given cograph can be computed in polynomial time. In this section, we generalize the result of Theorem 9.3.1 to prove the main result of this paper, a polynomial-time algorithm for computing a well-covering system of a given fork-free graph. This is a significant generalization of Theorem 9.3.1, since, more importantly, the class of fork-free graphs also generalizes the class of claw-free graphs. Our approach combines the results from Sections 9.2 and 9.4 with a known structural result on fork-free graphs, which allows us to reduce the problem to the class of claw-free graphs, for which the following theorem applies.

Theorem 9.5.1 (Levit and Tankus [164])**.** *There exists an $\mathcal{O}(n^3 m^{3/2})$ algorithm that receives as input a claw-free graph G with n vertices and $m \geq 1$ edges and computes a unit well-covering system of G.*

Following Remark 9.2.8 and the fact that the function f defined by the rule $f(n,m) = n^{\omega+2} m^{3/2}$ for all $m, n \geq 0$, is superadditive, Theorem 9.5.1 has the following consequence.

Corollary 9.5.2. *Let $\mathcal{C}$ be the class of all graphs G such that every prime induced subgraph of G is claw-free. Then for any graph G in $\mathcal{C}$ with n vertices and $m \geq 1$ edges, one can compute in time $\mathcal{O}\big(n^{\omega+2} m^{3/2}\big)$ a unit well-covering system of G with size at most n.*

To apply Corollary 9.5.2, we use the following structural result on fork-free graphs due to Lozin and Milanič [167, 168].[2]

Theorem 9.5.3. *Let G be a prime fork-free graph, let x be a vertex of G, and let G' be a prime induced subgraph of the graph $G - N[x]$. Then G' is claw-free.*

Using Theorems 9.4.3 and 9.5.3 and Corollary 9.5.2, we can now derive the following.

Lemma 9.5.4. *Given a prime fork-free graph G with n vertices and $m \geq 1$ edges, a unit well-covering system of G with size at most n can be computed in time $\mathcal{O}(n^{\omega+3}m^{3/2})$.*

Proof. Let $\mathcal{F}$ be the class of prime fork-free graphs and let $\mathcal{F}^*$ be the class of all graphs G such that every prime induced subgraph of G is claw-free. By Theorem 9.5.3, for every graph $G \in \mathcal{F}$ and every vertex $x \in V(G)$, the graph $G - N[x]$ belongs to $\mathcal{F}^*$. By Corollary 9.5.2, given a graph $G \in \mathcal{F}^*$ with n vertices and m edges one can compute in time $\mathcal{O}\big(n + n^{\omega+2}m^{3/2}\big)$ a unit well-covering system of G with size at most n, where the additive $\mathcal{O}(n)$ term has only been added in order to allow for G to be edgeless. Thus, by Theorem 9.4.3, given a graph $G \in \mathcal{F}$ with n vertices and $m \geq 1$ edges one can compute in time $\mathcal{O}\big(n \cdot \big(n + m + n^{\omega+2}m^{3/2}\big)\big) = \mathcal{O}\big(n^{\omega+3}m^{3/2}\big)$ a unit well-covering system $\widehat{\mathcal{S}}$ of G with size at most $n^2 + n - 1$. By Lemma 9.1.2, a unit well-covering subsystem $\mathcal{S} \subseteq \widehat{\mathcal{S}}$ of G, with size at most n, can be computed in time $\mathcal{O}(n^{\omega-1}|\widehat{\mathcal{S}}|) = \mathcal{O}(n^{\omega+1})$. The total time complexity of this approach is $\mathcal{O}\big(n^{\omega+3}m^{3/2}\big) + \mathcal{O}\big(n^{\omega+1}\big) = \mathcal{O}\big(n^{\omega+3}m^{3/2}\big)$, as claimed. □

We now have everything ready to prove the main result of the paper.

Theorem 9.5.5. *Given a fork-free graph G with n vertices and $m \geq 1$ edges, a unit well-covering system of G with size at most n can be computed in time $\mathcal{O}(n^{\omega+3}m^{3/2})$.*

Proof. Let $\mathcal{G}$ be the class of fork-free graphs and $\mathcal{G}^*$ the class of prime fork-free graphs. Lemma 9.5.4 implies that given a graph G in $\mathcal{G}^*$ with n vertices and $m \geq 1$ edges, a unit well-covering system of G with size at most n can be computed in time $\mathcal{O}(n^{\omega+3}m^{3/2})$. Let $f(n,m) = n^{\omega+3}m^{3/2}$. By Theorem 9.2.7, given a fork-free graph G with n vertices and $m \geq 1$ edges, a unit well-covering system $\mathcal{S}$ of G with size at most n can be computed in time $\mathcal{O}(f(2n,m) + n^{\omega+1}) = \mathcal{O}((2n)^{\omega+3}m^{3/2} + n^{\omega+1})$, which simplifies to $\mathcal{O}(n^{\omega+3}m^{3/2})$. □

We can determine if a graph G is well-covered by computing a well-covering system of G and checking if the weight function assigning 1 to each vertex of G satisfies all the equations in the system. This leads to the following consequence of Theorem 9.5.5.

Corollary 9.5.6. *There is a polynomial-time algorithm to determine if a given fork-free graph is well-covered.*

[2]The result is stated incorrectly in the paper [167]. It is stated correctly in the conference version of that work [168], as well as in the PhD thesis of the second-named author [182, Theorem 3.1.2]. The result was reproved by Dyer et al. in [97].

Chapter 10

Fair Allocation of Indivisible Items with Conflict Graphs

In this chapter we consider the fair allocation of indivisible items to several agents and add a graph theoretical perspective to this classical problem. In particular, we introduce an incompatibility relation between pairs of items described in terms of a conflict graph. Therefore, we study the FAIR k-DIVISION UNDER CONFLICTS that informally can be described as follows: given n items and k agents, where each agent has its own additive utility function over the set of items, and given a graph $G = (V, E)$ of incompatibilities among items, compute an allocation of items to agents so minimal utility over all agents is as large as possible (for a formal definition, see Section 10.1. We study the complexity of FAIR k-DIVISION UNDER CONFLICTS and our results can be summarized as follows:

(1) We show that the problem is strongly NP-hard for bipartite graphs and their line graphs.

(2) We show that the problem is solvable in pseudo-polynomial time for the classes of chordal graphs, cocomparability graphs, biconvex bipartite graphs, and graphs of bounded treewidth.

(3) We show that the problem is solvable in pseudo-polynomial time for the class of graphs of bounded clique-width.

The results (1) and (2) presented in this chapter are part of the following paper:

Chiarelli, N., Krnc, M., Milanič, M., Pferschy, U., Pivač, N., Schauer, J. Fair Allocation of Indivisible Items with Conflict Graphs. Algorithmica 85, 1459–1489 (2023). https://doi.org/10.1007/s00453-022-01079-8.

The result (3) is part of the following paper:

Chiarelli, N., Krnc, M., Milanič, M., Pferschy, U., Schauer, J. Fair allocation algorithms for indivisible items under structured conflict constraints. Comp. Appl. Math. 42, 302 (2023). https://doi.org/10.1007/s40314-023-02437-0.

Result (3) was obtained during the writing of the first paper, but was published within the second paper. Although I am not an author of the second paper, I have participated in the discussions that led to the result (3) and the authors of the paper agreed that this result should be part of my thesis and I thank them for that.

10.1 Introduction

We study the allocation of n indivisible goods or items to a set of k agents where each agent has its own profit function over the set of items. The aim is to assign every item to exactly one of the agents so that the allocation is *fair*, that is, the minimal total profit obtained by any of the agents should be maximized. This will be referred to as a *maxi-min* allocation. Additionally, we restrict the set of feasible solutions by introducing a notion of a *conflict graph*. A *conflict graph* is a graph G where $V(G)$ is the set of items and $E(G)$ expresses the incompatibilities. This might suggest that items restrict their shared usage, or it could be due to the fact that certain items are identical (or similar) in type, making it nonsensical for one agent to receive multiple copies of these items. In particular, given $u, v \in V(G)$, it holds that $uv \in E(G)$ if and only if u and v are incompatible items.

Each agent receives the set of items that are not pairwise incompatible, that is, every feasible allocation to one agent must be an independent set in the conflict graph, and in general, we allow the partial distribution of items, meaning that possibly not all items are distributed. Thus, the allocation of items to the agents corresponds to a partial k-coloring of the conflict graph, where a *partial k-coloring* of a graph G is a sequence $(X_1, \ldots, X_k)$ of k pairwise disjoint independent sets in G, see [22, 90]. In addition, every vertex/item has a profit value for every color/agent and the sum of profits of vertices/items assigned to one color/agent should be optimized in a maxi-min sense. Combining the profit structure with the notion of coloring we define for the k profit functions $p_1, \ldots, p_k : V \to \mathbb{Z}_+$ and for each partial k-coloring $c = (X_1, \ldots, X_k)$ a k-tuple $(p_1(X_1), \ldots, p_k(X_k))$, called the *profit profile* of c. The minimum profit of a profile, i.e., $\min_{j=1}^{k}\{p_j(X_j)\}$, is the *satisfaction level* of c.

We consider the following problem. In the hardness reductions of this paper we will frequently use the decision version of this problem: for a given $q \in \mathbb{Z}_+$, does there exist a partial k-coloring of G with satisfaction level at least q?

Fair k-Division Under Conflicts	
Instance:	A graph $G = (V, E)$, k profit functions $p_1, \ldots, p_k : V \to \mathbb{Z}_+$.
Question:	Compute a partial k-coloring of G with maximum satisfaction level.

Note that an optimal partial k-coloring $(X_1, \ldots, X_k)$ does not necessarily select all vertices from V. However if there are no conflicts in an instance (meaning that the graph G is edgeless) all vertices will be selected – this special case corresponds to Fair k-Division of Indivisible Items.

Observation 10.1.1. Fair k-Division of Indivisible Items, *even with k identical profit functions, is weakly* NP*-hard for any constant $k \geq 2$ and strongly* NP*-hard for k being part of the input.*

Note that for $k = 2$, the decision version of Fair k-Division of Indivisible Items also generalizes the decision version of the Knapsack problem: Given a set $V = \{1, \ldots, n\}$ of items with weights $w_1, \ldots, w_n \in \mathbb{Z}_+$ and values $v_1, \ldots, v_n \in \mathbb{Z}_+$, and two positive integers W and C such that $W < \sum_{j \in V} w_j$, is there a subset of the items having total weight at most W and total value at least C? [1]

[1] Indeed, by considering two profit functions $p_1, p_2 : V \to \mathbb{Z}_+$ defined by $p_1(i) = \Delta \cdot v_i$ where $\Delta = \sum_{j \in V} w_j - W$ and $p_2(i) = C \cdot w_i$ for all $i \in V$, it is not difficult to verify that such a set S exists if and only if V admits an ordered 2-partition with satisfaction level at least $C \cdot \Delta$.

It should be noted that FAIR k-DIVISION OF INDIVISIBLE ITEMS is still only weakly NP-hard for constant k even for arbitrary profit functions, since we can construct a pseudo-polynomial algorithm solving the problem with a k-dimensional dynamic programming array.

For $k = 1$, the problem coincides with the WEIGHTED INDEPENDENT SET problem: given a graph $G = (V, E)$ and a weight function on the vertices, find an independent set of maximum total weight. In particular, since the case of unit weights and $k = 1$ coincides with the INDEPENDENT SET problem, we obtain the following result.

Observation 10.1.2. FAIR 1-DIVISION UNDER CONFLICTS *is strongly* NP*-hard* .

Thus, the addition of the conflict structure gives rise to a much more complicated problem, since FAIR k-DIVISION OF INDIVISIBLE ITEMS (which arises naturally as a special case for an edgeless conflict graph G) is trivial for $k = 1$ and only weakly NP-hard for $k \geq 2$.

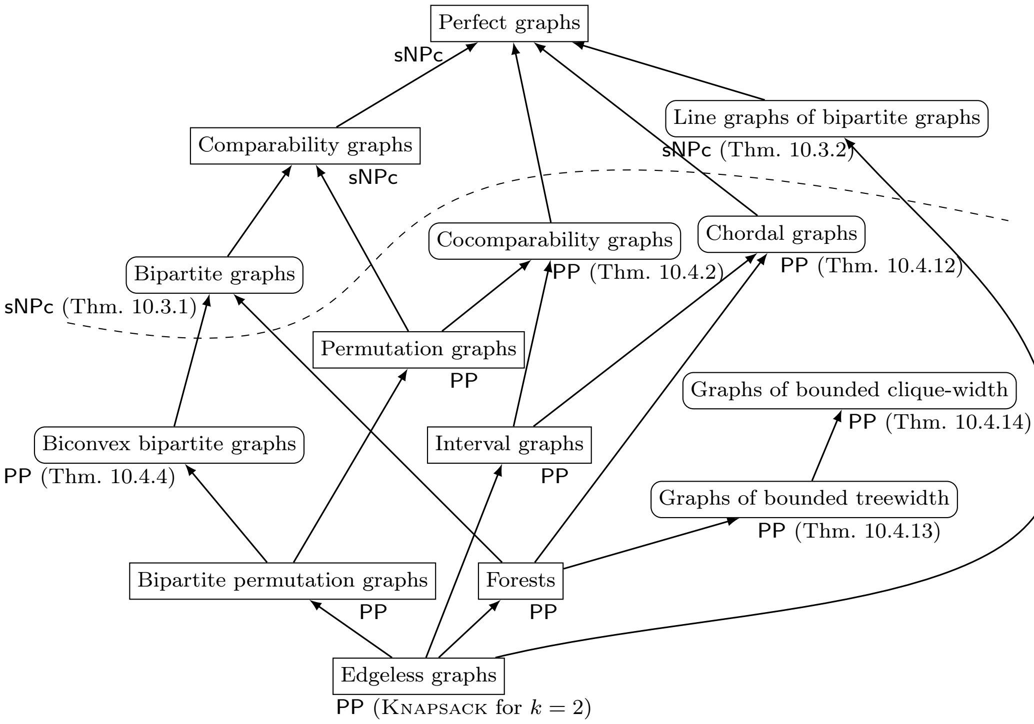


Figure 10.1: Relationships between various graph classes and the complexity of FAIR k-DIVISION UNDER CONFLICTS (decision version). An arrow from a class $\mathcal{G}_1$ to a class $\mathcal{G}_2$ means that every graph in $\mathcal{G}_1$ is also in $\mathcal{G}_2$. Label 'PP' means that for each fixed k the problem is solvable in pseudo-polynomial time in the given class, and label 'sNPc' means that for each fixed $k \geq 2$ the decision version of the problem is strongly NP-complete. For graph classes with round corners the result is shown in the cited theorem of this paper. Results depicted in rectangles follow from the inclusion of graph classes. For all graph classes in the figure, the problem is solvable in strongly polynomial time for $k = 1$, as it coincides with the WEIGHTED INDEPENDENT SET problem.

We give a characterization of the computational complexity of FAIR k-DIVISION UNDER CONFLICTS for different classes of conflict graphs and study the boundary between strongly NP-hard cases and those where a pseudo-polynomial algorithm can be derived for constant k. Observation 10.1.1 implies that this is the only type of positive result we can achieve. Moreover, considering Observation 10.1.2, it only makes sense to consider graph classes where the Weighted Independent Set problem is (pseudo-)polynomially solvable. One such prominent example is the class of perfect graphs (see [128]). Thus, we mainly concentrate on various subclasses of perfect graphs as depicted in Fig. 10.1. Additionally, we show how to adapt the algorithm for chordal graphs to obtain a pseudo-polynomial algorithm for graphs of bounded treewidth. For $k = 2$ our pseudo-polynomial dynamic programming approaches generalize the standard dynamic program for the Knapsack problem.

10.2 General hardness results

We start with the following general property of graph classes. Let us call a graph class $\mathcal{G}$ *sustainable* if every graph in the class can be enlarged in polynomial time to a graph in the class by adding to it one vertex. More formally, $\mathcal{G}$ is sustainable if there exists a polynomial-time algorithm that computes for every graph $G \in \mathcal{G}$ a graph $G' \in \mathcal{G}$ and a vertex $v \in V(G')$ such that $G' - v = G$. Clearly, any class of graphs closed under adding isolated vertices, or under adding universal vertices is sustainable. This property is shared by many well known graph classes, including planar graphs, bipartite graphs, chordal graphs, perfect graphs, etc. Furthermore, all graph classes defined by a single nontrivial forbidden induced subgraph are sustainable.

Lemma 10.2.1. *For every graph H with at least two vertices, the class of H-free graphs is sustainable.*

Proof. Let $\mathcal{G}$ be the class of H-free graphs and let $G \in \mathcal{G}$. Since H has at least two vertices, it cannot have both a universal and an isolated vertex. If H has no universal vertex, then the graph obtained from G by adding to it a universal vertex results in a graph in $\mathcal{G}$ properly extending G. If H has no isolated vertex, then the disjoint union of G with the one-vertex graph results in a graph in $\mathcal{G}$ properly extending G. □

For an example of a graph class $\mathcal{G}$ closed under vertex deletion that is not sustainable, consider the family of all cycles and their induced subgraphs. Then every cycle is in $\mathcal{G}$ but cannot be extended to a larger graph in $\mathcal{G}$. The importance of sustainable graph classes for Fair k-Division Under Conflicts is evident from the following theorem.

Theorem 10.2.2. *Let $\mathcal{G}$ be a sustainable class of graphs and let k be a positive integer such that the decision version of* Fair k-Division Under Conflicts *is (strongly)* NP*-complete. Then, for every $\ell \geq k$, the decision version of* Fair ℓ-Division Under Conflicts *with conflict graphs from $\mathcal{G}$ is (strongly)* NP*-complete.*

Proof. Let $\mathcal{G}$ be a sustainable class of graphs for which the decision version of Fair k-Division Under Conflicts is (strongly) NP-completeand let $\ell > k$. Let $(G, p_1, \ldots, p_k, q)$ be an instance of Fair k-Division Under Conflicts (decision version) such that $G \in \mathcal{G}$. Since $\mathcal{G}$ is sustainable, one can compute in polynomial time a graph $G' \in \mathcal{G}$ such that $G' - \{x_1, \ldots, x_{\ell-k}\} = G$ for some $\ell - k$ additional vertices $x_1, \ldots, x_{\ell-k}$. We now define the profit functions $p'_1, \ldots, p'_\ell : V(G') \to \mathbb{Z}_+$. For all $j = 1, \ldots, k$, let

$$p'_j(v) = \begin{cases} p_j(v) & \text{if } v \in V(G), \\ 0 & \text{if } v \in \{x_j \mid 1 \leq j \leq \ell - k\} . \end{cases}$$

and in addition let, for all $j = k+1, \ldots, \ell$, let

$$p_j(v) = \begin{cases} q & \text{if } v = x_{j-k}, \\ 0 & \text{if } v \in V(G') \setminus \{x_{j-k}\}. \end{cases}$$

Observe that G' has a partial k-coloring $(X'_1, \ldots, X'_k)$ such that $p'_j(X'_j) \geq q$ for all $j = 1, \ldots, \ell$ if and only if G has a partial k-coloring $(X_1, \ldots, X_k)$ such that $p_j(X_j) \geq q$ for all $j = 1, \ldots, k$. Since all the numbers involved in the reduction are polynomially bounded, we conclude that Fair ℓ-Division Under Conflicts with conflict graphs from $\mathcal{G}$ is also (strongly) NP-complete. □

Since the Independent Set problem is a special case of Fair 1-Division Under Conflicts, Theorem 10.2.2 immediately implies the following.

Corollary 10.2.3. *Let $\mathcal{G}$ be a sustainable class of graphs for which the decision version of* Independent Set *is* NP*-complete. Then, for every $k \geq 1$, the decision version of* Fair k-Division Under Conflicts *with conflict graphs from $\mathcal{G}$ is strongly* NP*-complete.*

It is known (see, e.g., [5]) that for every graph H that has a component that is not a path or a subdivision of the claw (the complete bipartite graph $K_{1,3}$), the decision version of Independent Set is NP-completeon H-free graphs. Thus, for every such graph H, Lemma 10.2.1 and Corollary 10.2.3 imply that for every $k \geq 1$, Fair k-Division Under Conflicts (decision version) with H-free conflict graphs is strongly NP-complete. Further exploiting the relation to Independent Set, we also get the following strong inapproximability result for general graphs. Its proof is closely related to the inapproximability result for Independent Set, but to keep the paper self-contained, we include it here.

Theorem 10.2.4. *For every $k \geq 1$ and every $\varepsilon > 0$, it is* NP*-hard to approximate* Fair k-Division Under Conflicts *within a factor of $|V(G)|^{1-\varepsilon}$, even for unit profit functions.*

Proof. Fix an integer $k \geq 1$. We give a reduction from the Independent Set problem. We construct a graph G' by taking k copies of G and by adding all possible edges between vertices from different copies. Furthermore we take k "unit" profit functions $p_1, \ldots, p_k$ from $V(G')$ to $\{1\}$. We claim that the maximum size of an independent set in G equals the maximum satisfaction level of a partial k-coloring in G' (with respect to the profit functions $p_1, \ldots, p_k$). Given a maximum independent set I in G of size q one can immediately obtain a partial k-coloring $(X_1, \ldots, X_k)$ of G' with satisfaction level q by inserting all vertices of I in the j-th copy of G into X_j, for all $j = 1, \ldots, k$. On the other hand, given a partial k-coloring $(X_1, \ldots, X_k)$ of G' with satisfaction level q, one can simply choose X_1, which is an independent set completely contained in one copy of G. Thus, X_1 corresponds to an independent set in G of size q.

Here is a more detailed argument: We claim that the maximum size of an independent set in G equals the maximum satisfaction level of a partial k-coloring in G' (with respect to the unit profit functions $p_1, \ldots, p_k$). On the one hand, if I is an independent set of size q in G, then a partial k-coloring $(X_1, \ldots, X_k)$ of G' with satisfaction level q can be obtained by taking X_j to be a copy of I corresponding to the j-th copy of G in G'. In particular, this shows that the maximum satisfaction level of a partial k-coloring in G' is at least as large as the independence number of G. For the reverse direction, consider a partial k-coloring $(X_1, \ldots, X_k)$ of G' with satisfaction level q. Since X_1 is an independent set in G' and every two vertices of G' belonging to different copies of G are adjacent, X_1 must be contained in a single copy of G. Thus, X_1 corresponds to an independent set of size at least q in G. This shows that the independence number of G is at least the maximum satisfaction level of a partial k-coloring in G'.

Suppose that for some $\varepsilon \in (0,1)$ there exists a polynomial-time algorithm A that approximates Fair k-Division Under Conflicts within a factor of $|V(G)|^{1-\varepsilon}$ on input instances with unit profit functions. We will show that this implies the existence of a polynomial-time algorithm A' approximating the Independent Set problem within a factor of $|V(G)|^{1-\varepsilon'}$ where $\varepsilon' = \varepsilon/2$. As shown by Zuckerman [234], this would imply $\mathsf{P} = \mathsf{NP}$.

Consider an input graph G to the Independent Set problem. The algorithm A' proceeds as follows. If $|V(G)| < k^{2(1-\varepsilon)/\varepsilon}$, then the graph is of constant order and the problem can be solved optimally in $\mathcal{O}(1)$ time. If $|V(G)| \geq k^{2(1-\varepsilon)/\varepsilon}$, then the graph G' is constructed following the above reduction, a partial k-coloring $(X_1, \ldots, X_k)$ is computed using algorithm A on G' equipped with k unit profit functions, and a subset of $V(G)$ corresponding to X_1 is returned. Clearly, the algorithm runs in polynomial time and computes an independent set in G. Let q denote the maximum satisfaction level of a partial k-coloring in G'. By the above claim, the independence number of G equals q. Thus, to complete the proof, it suffices to show that $|X_1| \geq q/(|V(G)|^{1-\varepsilon'})$. By assumption on A, we have that $|X_1| \geq q/(|V(G')|^{1-\varepsilon})$. We want to show that $q/|V(G')|^{1-\varepsilon} \geq q/|V(G)|^{1-\varepsilon'}$, or, equivalently, $1/k^{1-\varepsilon}|V(G)|^{1-\varepsilon} \geq 1/|V(G)|^{1-\varepsilon/2}$. After some straightforward algebraic manipulations, this inequality simplifies to the equivalent inequality $|V(G)| \geq k^{2(1-\varepsilon)/\varepsilon}$, which is true by assumption. $\square$

10.3 Bipartite graphs and their line graphs

In this section we show that for all $k \geq 2$, Fair k-Division Under Conflicts is NP-hard in two classes of graphs where the Weighted Independent Set problem is solvable in polynomial time: the classes of bipartite graphs and their line graphs. Recall that for a graph H, its line graph has a vertex for each edge of H, with two distinct vertices adjacent in the line graph if and only if the corresponding edges share an endpoint in H. Polynomial-time solvability of the Weighted Independent Set problem in the class of bipartite graphs is well-known from a reduction to a network flow problem (see, e.g., [141] or [216, Corollary 21.25a]). For line graphs of bipartite graphs polynomial-time solvability follows from the facts that we can compute in linear time a bipartite graph H such that the input graph G is the line graph of H [162, 213] and that the Weighted Independent Set problem on G is equivalent to the weighted matching problem on H. Clearly, polynomial-time solvability for the two classes also follows from the fact that both classes are subclasses of the class of perfect graphs (cf. Fig. 10.1 and [216, Section 66.1]).

The proof for bipartite graphs shows strong NP-hardness even for the case when all the profit functions are equal.

Theorem 10.3.1. *For each integer $k \geq 2$, the decision version of* Fair k-Division Under Conflicts *is strongly* NP*-complete in the class of bipartite graphs.*

Proof. We use a reduction from the decision version of the CLIQUE problem: Given a graph G and an integer ℓ, does G contain a clique of size ℓ? Consider an instance (G, ℓ) of CLIQUE such that $2 \leq \ell < n := |V(G)|$. We define an instance of FAIR k-DIVISION UNDER CONFLICTS (decision version) consisting of a bipartite conflict graph G', profit functions $p_1, \ldots, p_k$, and a lower bound q on the required satisfaction level. The graph $G' = (A \cup B, E')$ has a vertex for each vertex of the graph G as well as for each edge of G and k new vertices $x_1, \ldots, x_k$. It is defined as follows:

$$\begin{aligned} A &= V(G) \cup \{x_1\}\,, \; B = E(G) \cup \{x_i \mid 2 \leq i \leq k\}\,, \\ E' &= \{ve \mid v \in V(G) \text{ is an endpoint of } e \in E(G)\} \cup \{vx_i \mid v \in V(G), 2 \leq i \leq k\}\,. \end{aligned}$$

The lower bound q on the satisfaction level is defined by setting $q = n^4 + \binom{\ell}{2}n + (n - \ell)$. For ease of notation we set $N_1 = n^4$ and we furthermore introduce a second integer N_2 such that $q = N_2 + \left(m - \binom{\ell}{2}\right) n$, where $m = |E(G)|$. (Note that $N_2 \geq n^3$.) With this, the profit functions $p_i : V(G') \to \mathbb{Z}_+$, for all $i \in \{1, \ldots, k\}$, are defined as

$$p_i(v) = \begin{cases} 1; & \text{if } v \in V(G); \\ n; & \text{if } v \in E(G); \\ N_1; & \text{if } v = x_1; \\ N_2; & \text{if } v = x_2; \\ q; & \text{if } v = x_j \text{ for some } j \in \{3, \ldots, k\}. \end{cases}$$

Note that all the profits introduced as well as the number of vertices and edges of G' are polynomial in n. To complete the proof, we show that G has a clique of size ℓ if and only if G' has a partial k-coloring with satisfaction level at least q. First assume that G has a clique C of size ℓ. We construct a partial k-coloring $c = (X_1, \ldots, X_k)$ of G' by setting

$$\begin{aligned} X_1 &= \{x_1\} \cup \{e \in E(G) \mid e \subseteq C\} \cup (V(G) \setminus C)\,, \\ X_2 &= \{x_2\} \cup (E(G) \setminus X_1)\,, \\ X_j &= \{x_j\} \;\text{ for } 3 \leq j \leq k. \end{aligned}$$

Observe that the partial k-coloring c gives rise to the corresponding profit profile with all entries equal to q, which establishes one of the two implications.

Suppose now that there exists a partial k-coloring $c = (X_1, \ldots, X_k)$ of G' for which the profit profile has all entries $\geq q$. Since for each $i \in \{1, \ldots, k\}$, the total profit of the set $V(G) \cup E(G)$ is only $mn + n < n^4$, the partial coloring c must use exactly one of the k vertices $x_1, \ldots, x_k$ in each color class. We may assume without loss of generality that $x_i \in X_i$ for all $i \in \{1, \ldots, k\}$. Let U be the set of uncolored vertices in G' w.r.t. the partial coloring c. Since for each of the profit functions p_i, the difference between the overall sum of the profits of vertices of G' and $k \cdot q$ is equal to ℓ, we clearly have $\sum_{v \in U} p_i(v) \leq \ell < n$, which implies that $U \subseteq V(G)$. Next, observe that every vertex of $E(G)$ belongs to either X_1 or to X_2, since otherwise we would have $p_1(X_1) + p_2(X_2) < 2q$, contrary to the assumption that the satisfaction level of c is at least q.

Consider the sets $W = X_1 \cap V(G)$ and $F = X_1 \cap E(G)$. Then $X_1 = \{x_1\} \cup W \cup F$ and, since $\sum_{v \in X_1} p_1(v) \geq q = N_1 + \binom{\ell}{2}n + (n - \ell)$, it follows that X_1 contains exactly $\binom{\ell}{2}$ vertices from $E(G)$ (if $|F| > \binom{\ell}{2}$, then $p_2(X_2) < q$) and at least $n - \ell$ vertices from $V(G)$. Let C denote the set of all vertices of G' with a neighbor in F. By the construction of

G' and since $|F| = \binom{\ell}{2}$, it follows that C is of cardinality at least ℓ. Furthermore, since X_1 is independent, we have $C \cap W = \emptyset$. Consequently, $n = |V(G)| \geq |C| + |W| \geq \ell + (n - \ell) = n$, hence equalities must hold throughout. In particular, C is a clique of size ℓ in G. □

Theorem 10.3.2. *For each integer $k \geq 2$, the decision version of* Fair k-Division Under Conflicts *is strongly* NP*-complete in the class of line graphs of bipartite graphs.*

Proof. Note that it suffices to prove the statement for $k = 2$. For $k > 2$, Theorem 10.2.2 applies, since the class of line graphs of bipartite graphs is sustainable. Indeed, if G' is the line graph of a bipartite graph G, then the graph obtained from G' by adding to it an isolated vertex is the line graph of the bipartite graph obtained from G by adding to it an isolated edge.

For $k = 2$, we use a reduction from the following problem: Given a bipartite graph G and an integer Q, does G contain two disjoint matchings M_1 and M_2 such that M_1 is a perfect matching and $|M_2| \geq Q$? This problem was shown to be NP-completeby Pálvölgi (see [192]). Consider an instance (G, Q) of this problem such that $1 \leq Q \leq n/2$ and $n = |V(G)|$ is even. Then we define the following instance of the decision version of Fair 2-Division Under Conflicts with a conflict graph G', where G' is the line graph of G. The lower bound q on the satisfaction level is defined by setting $q = n \cdot Q/2$. The profit functions $p_1, p_2 : V(G') \to \mathbb{Z}_+$ are defined as $p_1(v) = Q$ for all $v \in V(G')$, and $p_2(v) = n/2$ for all $v \in V(G')$. Clearly, all the profits introduced as well as the number of vertices and edges of G' are polynomial in n. Recall that every matching in G corresponds to an independent set in G'.

We now show that the instances of the two decision problems have the same answers. Suppose first that G has two disjoint matchings M_1 and M_2 such that M_1 is a perfect matching and $|M_2| \geq Q$. Then the sequence (M_1, M_2) is a partial 2-coloring of G' such that

$$p_1(M_1) = Q|M_1| = Q \cdot n/2 = q \text{ and } p_2(M_2) = (n/2) \cdot |M_2| \geq (n/2)Q = q.$$

Conversely, suppose that G' has a partial 2-coloring (X_1, X_2) with satisfaction level at least q. Then the independent sets X_1 and X_2 in G' are disjoint matchings in G. Moreover, since

$$p_1(X_1) = Q|X_1| \geq q = Q \cdot n/2 \text{ and } p_2(X_2) = (n/2) \cdot |X_2| \geq q = Q \cdot n/2,$$

we obtain $|X_1| \geq n/2$ and $|X_2| \geq Q$. Thus, X_1 is a perfect matching in G and any set of Q edges in X_2 is a matching in G disjoint from X_1. This proves that the decision version of Fair 2-Division Under Conflicts is strongly NP-complete in the class of line graphs of bipartite graphs. □

10.4 Pseudo-polynomial algorithms for special graph classes

In this section we turn our attention to classes of graphs for which the Fair k-Division Under Conflicts is solvable in pseudo-polynomial time. As shown in Theorem 10.3.1, for each $k \geq 2$, Fair k-Division Under Conflicts is strongly

NP-complete in the class of bipartite graphs, and this rules out the existence of a pseudo-polynomial time algorithm for the problem in the class of bipartite graphs, unless P = NP. We show that for every k there is a pseudo-polynomial time algorithm for the FAIR k-DIVISION UNDER CONFLICTS in a subclass of bipartite graphs, the class of *biconvex bipartite graphs* (see the definition in Section 10.4.2). The algorithm reduces the problem to the class of bipartite permutation graphs. To solve the problem in the class of bipartite permutation graphs, we develop a solution in a more general class of graphs, the class of cocomparability graphs (containing permutation graphs). Further, using a dynamic programming approach, we show that for every k there is a pseudo-polynomial time algorithm for FAIR k-DIVISION UNDER CONFLICTS in the classes of chordal graphs and graphs of bounded treewidth. It will be shown in Section 10.5 that all these pseudo-polynomial dynamic programming algorithms allow the construction of a fully polynomial time approximation scheme (FPTAS).

Let us first fix some notation. Given a graph G and k profit functions $p_1, \ldots, p_k : V \to \mathbb{Z}_+$, we denote by n the number of vertices in G, $n = |V(G)|$. All pseudo-polynomial results in this section depend on an upper bound on the maximum reachable profit value $Q = \max_{1 \le j \le k} p_j(V)$. Given an integer $k > 0$, the addition and subtraction of k-tuples is defined component-wise, and for all $\ell \in \{1, \ldots, k\}$, we denote by $\mathbf{e}_\ell(x)$ the k-tuple with all coordinates equal to 0, except that the ℓ-th coordinate is equal to x.

10.4.1 Cocomparability graphs

A graph $G = (V, E)$ is a *comparability graph* if it has a transitive orientation, that is, if each of the edges $\{u, v\}$ of G can be replaced by exactly one of the ordered pairs (u, v) and (v, u) so that the resulting set A of directed edges is transitive (that is, for every three vertices $x, y, z \in V$, if $(x, y) \in A$ and $(y, z) \in A$, then $(x, z) \in A$). A graph G is a *cocomparability graph* if its complement is a comparability graph. Comparability graphs and cocomparability graphs are well-known subclasses of perfect graphs. The class of cocomparability graphs is a common generalization of the classes of interval graphs, permutation graphs, and trapezoid graphs (see, e.g., [46, 127]).

Since every bipartite graph is a comparability graph, Theorem 10.3.1 implies that for each $k \ge 2$, FAIR k-DIVISION UNDER CONFLICTS is strongly NP-complete in the class of comparability graphs. For cocomparability graphs, we prove that the problem is solvable in pseudo-polynomial time. The key result in this direction is the following lemma.

Lemma 10.4.1. *For every $k \ge 1$, given a cocomparability graph $G = (V, E)$ and k profit functions $p_1, \ldots, p_k : V \to \mathbb{Z}_+$, the set of all profit profiles of partial k-colorings of G can be computed in time $\mathcal{O}(n^{k+2}(Q+1)^k)$, where $Q = \max_{1 \le j \le k} p_j(V)$.*

Proof. Let G be a cocomparability graph. In time $\mathcal{O}(n^2)$, we compute the complement of G and a transitive orientation D of it [219]. Since D is a directed acyclic graph, one can compute in linear time a topological sort of D, that is, an ordering $v_1, \ldots, v_n$ of the vertices such that if (v_i, v_j) is an arc of D, then $i < j$ (see, e.g., [71]). Note that

(∗) a set $X = \{v_{i_1}, \ldots, v_{i_p}\} \subseteq V$ with $i_1 < \ldots < i_p$ is independent in G if and only if $(v_{i_1}, \ldots, v_{i_p})$ is a directed path in D.

Thus, a partial k-coloring in G corresponds to a collection of k vertex-disjoint directed paths in D, and vice versa. We process the vertices of G in the ordering given by the topological sort of D and try all possibilities for the color (if any) of the current vertex v_j in order to extend a partial k-coloring of the already processed subgraph of G with v_j. (In terms of D, we choose which of the k directed paths will be extended into v_j.) To avoid introducing additional terminology and notation, we present the details of the algorithm in terms of partial k-colorings of G instead of systems of disjoint paths in D.

For each $j \in \{0, 1, \ldots, n\}$ and each k-tuple $(i_1, \ldots, i_k) \in \{0, 1, \ldots, j\}^k$, we compute the set $P_j(i_1, \ldots, i_k)$ of all k-tuples $(q_1, \ldots, q_k) \in \mathbb{Z}_+^k$ such that there exists a partial k-coloring $(X_1, \ldots, X_k)$ of the subgraph of G induced by $\{v_1, \ldots, v_j\}$ (which is empty if $j = 0$) such that $q_\ell = p_\ell(X_\ell)$ and

$$i_\ell = \begin{cases} \max\{r : v_r \in X_\ell\}, & \text{if } X_\ell \neq \emptyset; \\ 0, & \text{if } X_\ell = \emptyset \end{cases} \tag{10.1}$$

for all $\ell \in \{1, \ldots, k\}$. Note that for each $\ell \in \{1, \ldots, k\}$, the possible values of the ℓ-th coordinate of any member of $P_j(i_1, \ldots, i_k)$ belong to the set $\{0, 1, \ldots, Q\}$ where $Q = \max_{1 \le j \le k} p_j(V)$. Thus, each set $P_j(i_1, \ldots, i_k)$ has at most $(Q+1)^k$ elements. Note also that the total number of sets $P_j(i_1, \ldots, i_k)$ is of the order $\mathcal{O}(n^{k+1})$.

In what follows we explain how to compute the sets $P_j(i_1, \ldots, i_k)$. For $j = 0$, the only feasible choice for the k-tuple $(i_1, \ldots, i_k)$ is $(0, \ldots, 0)$ and we set $P_0(0, \ldots, 0) = \{0\}^k = \{(0, \ldots, 0)\}$. This is correct since the only partial k-coloring of the graph with no vertices is the k-tuple $(\emptyset, \ldots, \emptyset)$. Suppose that $j > 1$ and that the sets $P_{j-1}(i_1, \ldots, i_k)$ are already computed for all $(i_1, \ldots, i_k) \in \{0, 1, \ldots, j-1\}^k$. Fix a k-tuple $(i_1, \ldots, i_k) \in \{0, 1, \ldots, j\}^k$. To describe how to compute the set $P_j(i_1, \ldots, i_k)$, we will use the following notation. We consider three cases. For each of them, we first give a formula for computing the set $P_j(i_1, \ldots, i_k)$ and then we argue why the formula is correct.

1. If j appears at least twice as a coordinate of $(i_1, \ldots, i_k)$, then we set

$$P_j(i_1, \ldots, i_k) = \emptyset\,. \tag{10.2}$$

   Note that since j appears at least twice as a coordinate of $(i_1, \ldots, i_k)$, there is no partial k-coloring $(X_1, \ldots, X_k)$ of the subgraph of G induced by $\{v_1, \ldots, v_j\}$ such that equality (10.1) holds for all $\ell \in \{1, \ldots, k\}$. Thus, equation (10.2) is correct.

2. If j does not appear as any coordinate of $(i_1, \ldots, i_k)$, then we set

$$P_j(i_1, \ldots, i_k) = P_{j-1}(i_1, \ldots, i_k)\,. \tag{10.3}$$

   Since j does not appear as any coordinate of $(i_1, \ldots, i_k)$, every partial k-coloring of the subgraph of G induced by $\{v_1, \ldots, v_{j-1}\}$ such that equality (10.1) holds for all $\ell \in \{1, \ldots, k\}$ is a partial k-coloring of the subgraph of G induced by $\{v_1, \ldots, v_j\}$ and vice versa. This implies relation (10.3).

3. If j appears exactly once as a coordinate of $(i_1, \ldots, i_k)$, say $i_s = j$, then we set

$$P_j(i_1, \ldots, i_k) = \bigcup_{\substack{\{j' : j'=0 \text{ or} \\ v_{j'} \in N_D^-(v_j)\}}} \{\mathbf{q} + \mathbf{e}_s(p_s(v_j)) \mid \mathbf{q} \in P_{j-1}(i_1, \ldots, i_{s-1}, j', i_{s+1}, \ldots, i_k)\}\,, \tag{10.4}$$

where $N_D^-(v_j)$ denotes the set of all vertices $v_{j'}$ such that $(v_{j'}, v_j)$ is an arc of D. (Note that $j' < j$ for all $v_{j'} \in N_D^-(v_j)$, since $v_1, \ldots, v_n$ is a topological sort of D.)

Let $\mathbf{q} = (q_1, \ldots, q_k) \in P_j(i_1, \ldots, i_k)$ and consider a partial k-coloring $(X_1, \ldots, X_k)$ of the subgraph of G induced by $\{v_1, \ldots, v_j\}$ such that $p_\ell(X_\ell) = q_\ell$ and equality (10.1) holds for all $\ell \in \{1, \ldots, k\}$. Then $\max\{q : v_q \in X_s\} = i_s = j$. In particular, $v_j \in X_s$. Let $X_s' = X_s \setminus \{v_j\}$ and let

$$j' = \begin{cases} \max\{r : v_r \in X_s'\}, & \text{if } X_s' \neq \emptyset; \\ 0, & \text{if } X_s' = \emptyset. \end{cases}$$

Note that if $X_s' \neq \emptyset$ then $v_{j'} \in N_D^-(v_j)$. Indeed, digraph D is an orientation of the complement of G, in which vertices $v_{j'}$ and v_j are adjacent (recall that they belong to the independent set X_s in G). This implies that either $(v_j, v_{j'})$ or $(v_{j'}, v_j)$ is an arc of D, but since $j' < j$ and $v_1, \ldots, v_n$ is a topological sort of D, the pair $(v_{j'}, v_j)$ must be an arc of D. Let $(i_1', \ldots, i_k')$ be the k-tuple obtained from $(i_1, \ldots, i_k)$ by replacing i_s with j', and let $(X_1', \ldots, X_k')$ be the k-tuple obtained from $(X_1, \ldots, X_k)$ by replacing X_s with X_s'. Then $(X_1', \ldots, X_k')$ is a partial k-coloring of the subgraph of G induced by $\{v_1, \ldots, v_{j-1}\}$ such that equality obtained from (10.1) by replacing X_ℓ with X_ℓ' and i_ℓ with i_ℓ' holds for each $\ell \in \{1, \ldots, k\}$. Furthermore, $(p_1(X_1), \ldots, p_k(X_k)) = (p_1(X_1'), \ldots, p_k(X_k')) + \mathbf{e}_s(p_s(v_j))$. This shows that if $\mathbf{q} = (q_1, \ldots, q_k) \in P_j(i_1, \ldots, i_k)$, then the k-tuple $\mathbf{q}$ belongs to the union

$$\bigcup_{\{j' : j'=0 \text{ or } v_{j'} \in N_D^-(v_j)\}} \{\mathbf{q} + \mathbf{e}_s(p_s(v_j)) \mid \mathbf{q} \in P_{j-1}(i_1, \ldots, i_{s-1}, j', i_{s+1}, \ldots, i_k)\}\,.$$

For the converse direction, let $j' \in \{0\} \cup \{1 \leq j' \leq j-1 \mid v_{j'} \in N_D^-(v_j)\}$, let $(i_1', \ldots, i_k')$ be the k-tuple obtained from $(i_1, \ldots, i_k)$ by replacing i_s with j', and let $\mathbf{q} = (q_1, \ldots, q_k) \in P_{j-1}(i_1', \ldots, i_k')$. Then, there exists a partial k-coloring $(X_1', \ldots, X_k')$ of the subgraph of G induced by $\{v_1, \ldots, v_{j-1}\}$ such that for each $\ell \in \{1, \ldots, k\}$, we have $p_\ell(X_\ell') = q_\ell$ and equality obtained from (10.1) by replacing X_ℓ with X_ℓ' and i_ℓ with i_ℓ' holds. Let $(X_1, \ldots, X_k)$ be the k-tuple obtained from $(X_1', \ldots, X_k')$ by replacing X_s' with $X_s' \cup \{v_j\}$. To show that $(X_1, \ldots, X_k)$ is a partial k-coloring of the subgraph of G induced by $\{v_1, \ldots, v_j\}$, it suffices to verify that $X_s = X_s' \cup \{v_j\}$ is an independent set in G. If $X_s' = \emptyset$, then $X_s = \{v_j\}$ is independent. Suppose that $X_s' \neq \emptyset$. Then, by $(*)$, X_s' corresponds to a directed path in D ending in $v_{j'}$. Extending this path with vertex $v_j \in N_D^+(v_{j'})$ results in a directed path in D with vertex set X_s, which shows, again by $(*)$, that X_s is independent in G. Clearly, we have that $\max\{r : v_r \in X_s\} = j$, and hence $(X_1, \ldots, X_k)$ is a partial k-coloring of the subgraph of G induced by $\{v_1, \ldots, v_j\}$ equality (10.1) holds for each $\ell \in \{1, \ldots, k\}$. Furthermore, $(p_1(X_1), \ldots, p_k(X_k)) = \mathbf{q} + \mathbf{e}_s(p_s(v_j))$. This shows that if $\mathbf{q} \in P_{j-1}(i_1', \ldots, i_k')$, then the k-tuple $\mathbf{q}+\mathbf{e}_s(p_s(v_j))$ belongs to $P_j(i_1, \ldots, i_k)$. Therefore, equation (10.4) is correct.

Finally, the set of all profit profiles of partial k-colorings of G equals to the union, over all $(i_1, \ldots, i_k) \in \{0, 1, \ldots, n\}^k$, of the sets $P_n(i_1, \ldots, i_k)$.

The algorithm can be easily modified so that for each profit profile also a corresponding partial k-coloring is computed. We would just need to store, for each

$j \in \{0, 1, \ldots, n\}$, each $(i_1, \ldots, i_k) \in \{0, 1, \ldots, j\}^k$, and each k-tuple $(q_1, \ldots, q_k) \in P_j(i_1, \ldots, i_k)$, one partial k-coloring $(X_1, \ldots, X_k)$ of the subgraph of G induced by $\{v_1, \ldots, v_i\}$ such that $p_\ell(X_\ell) = q_\ell$ and equality (10.1) holds for all $\ell \in \{1, \ldots, k\}$.

It remains to estimate the time complexity of the algorithm. For each $j \in \{1, \ldots, n\}$ and each of the $\mathcal{O}(n^k)$ k-tuples $(i_1, \ldots, i_k) \in \{0, 1, \ldots, j\}^k$, we can decide which of the three cases (i)–(iii) occurs in time $\mathcal{O}(k)$. Step (10.2) takes constant time, step (10.3) takes time $\mathcal{O}((Q+1)^k)$, and step (10.4) can be implemented in time $\mathcal{O}(n(Q+1)^k)$. Altogether, this results in running time $\mathcal{O}(n(Q+1)^k)$ for each fixed $j \in \{1, \ldots, n\}$ and each k-tuple $(i_1, \ldots, i_k) \in \{0, 1, \ldots, j\}^k$. Consequently, the total running time of the algorithm is $\mathcal{O}(n^{k+2}(Q+1)^k)$. □

Lemma 10.4.1 implies the following.

Theorem 10.4.2. *For every $k \geq 1$,* Fair k-Division Under Conflicts *is solvable in time $\mathcal{O}(n^{k+2}(Q+1)^k)$ for cocomparability conflict graphs G, where $Q = \max_{1 \leq j \leq k} p_j(V(G))$.*

Proof. By Lemma 10.4.1, we can compute the set Π of all profit profiles of partial k-colorings of G in the stated running time. For each profit profile in Π, we can determine the satisfaction level of the corresponding partial k-coloring of G. Taking the maximum satisfaction level over all profiles gives the optimal value of Fair k-Division Under Conflicts for $(G, p_1, \ldots, p_k)$. □

10.4.2 Biconvex bipartite graphs

Recall from Theorem 10.3.1 that Fair k-Division Under Conflicts is strongly NP-hard for bipartite conflict graphs. Thus, we consider in the following the more restricted case of *biconvex* bipartite conflict graphs. Recall that a bipartite graph $G = (A \cup B, E)$ is biconvex if it has a *biconvex ordering*, that is, an ordering of A and B such that for every vertex $a \in A$ (resp. $b \in B$) the neighborhood $N(a)$ (resp. $N(b)$) is an interval of consecutive vertices in the ordering of B (resp. ordering of A).

It is known that a connected biconvex bipartite graph G can always be ordered in such a way that the first and last vertices on one side have a special structure. Fix a biconvex ordering of G, say $A = (a_1, \ldots, a_s)$ and $B = (b_1, \ldots, b_t)$. Define a_L (resp. a_R) as the vertex in $N(b_1)$ (resp. $N(b_t)$) whose neighborhood is not properly contained in any other neighborhood set (see [1, Def. 8]). In case of ties, a_L is the smallest such index (and a_R the largest). We always assume that $a_L \leq a_R$, otherwise the ordering in A could be mirrored. Under these assumptions, the neighborhoods of vertices appearing in the ordering before a_L and after a_R are nested.

Lemma 10.4.3 (Abbas and Stewart [1]). *Let $G = (A \cup B, E)$ be a connected biconvex graph. Then there exists a biconvex ordering of the vertices of G such that:*

1. *For all a_i, a_j with $a_1 \leq a_i < a_j \leq a_L$ we have $N(a_i) \subseteq N(a_j)$.*
2. *For all a_i, a_j with $a_R \leq a_i < a_j \leq a_s$ we have $N(a_j) \subseteq N(a_i)$.*
3. *The subgraph G' of G induced by vertex set $\{a_L, \ldots, a_R\} \cup B$ is a bipartite permutation graph.*

Property (iii) can be put in context with Theorem 10.4.2. Indeed, it is known that every permutation graph is a cocomparability graph (see, e.g., [46]). This gives rise to the following result that FAIR k-DIVISION UNDER CONFLICTS on biconvex bipartite graphs is indeed easier (from the complexity point of view) than on general bipartite graphs. The high-level idea of the algorithm is illustrated in Algorithm 1.

Algorithm 1 Algorithmic Idea for a Connected Biconvex Graph G

apply Lemma 10.4.3 for getting the cocomparability graph G' and vertices a_L, a_R
let $A_L := \{a_1, \ldots, a_{L-1}\}$ and $A_R := \{a_{R+1}, \ldots, a_s\}$
for all $j \in \{1, \ldots, k\}$ **do**
  guess $\overline{a}_j \in A_L$ with largest index (resp. smallest index $\underline{a}_j \in A_R$) included in X_j
end for
each such guess can be represented by a $2k$-tuple $\sigma = (\overline{a}_1, \ldots, \overline{a}_k, \underline{a}_1, \ldots, \underline{a}_k)$
for each guess σ **do**
  for all $j \in \{1, \ldots, k\}$ **do**
    exclude all vertices v of the neighborhood $N(\overline{a}_j) \subseteq B$ (and $N(\underline{a}_j) \subseteq B$)
    from insertion into X_j by setting their profit $p_j(v) := 0$
  end for
  apply Lemma 10.4.1 to the cocomparability graph G' and the modified profit functions to obtain the set Π_σ of all profit profiles $(q_1, \ldots, q_k)$ of partial k-colorings of G' with respect to the modified profits
  increase each profit profile by setting $q_j := q_j + p_j(\overline{a}_j) + p_j(\underline{a}_j)$
  augment these profiles with vertices from A_L and A_R
end for
choose the best solution over all guesses σ

Theorem 10.4.4. *For every $k \geq 1$,* FAIR k-DIVISION UNDER CONFLICTS *is solvable in time $\mathcal{O}(n^{3k+2}(Q+1)^k)$ for connected biconvex bipartite conflict graphs G, where $Q = \max_{1 \leq j \leq k} p_j(V(G))$.*

Proof. At first Lemma 10.4.3 is applied for obtaining from G the cocomparability graph G'. However, we have to consider also the vertex sets $A_L := \{a_1, \ldots, a_{L-1}\}$ and $A_R := \{a_{R+1}, \ldots, a_s\}$. This is done by considering assignments of vertices in $A_L \cup A_R$ to the k subsets of a partial k-coloring of G in an efficient way as follows.

For every $j \in \{1, \ldots, k\}$, we guess, by going through all possibilities, the largest index vertex $\overline{a}_j \in A_L$ (resp. smallest index $\underline{a}_j \in A_R$) inserted in X_j. One can add an artificial vertex a_0 (resp. a_{s+1}) to represent the case that no vertex from A_L (resp. A_R) is inserted in X_j. Thus, every guess is represented by a $2k$-tuple $\sigma = (\overline{a}_1, \ldots, \overline{a}_k, \underline{a}_1, \ldots, \underline{a}_k)$. The total number of such guesses (i.e., iterations) is bounded by $(n+1)^k$ for each of A_L and A_R, i.e., $\mathcal{O}(n^{2k})$ selections to be considered in total.

For each such guess σ we perform the following computations. For every $j \in \{1, \ldots, k\}$ the vertices in the neighborhood $N(\overline{a}_j) \subseteq B$ (and $N(\underline{a}_j) \subseteq B$) of the chosen index must be excluded from insertion into the corresponding set X_j. This can be easily realized by setting to 0 the profits p_j of all vertices in $N(\overline{a}_j)$ (resp. $N(\underline{a}_j)$). With these slight modifications of the profits we can apply Lemma 10.4.1 for the cocomparability graph G' and the modified profit functions p_j^σ to obtain the set Π_σ of all (pseudo-polynomially many) profit profiles $(q_1, \ldots, q_k)$ of partial k-colorings of G' with respect to p^σ. Every entry q_j of a profit profile in Π_σ is increased by $p_j(\overline{a}_j)+p_j(\underline{a}_j)$, to account for inclusion of the vertices selected by the guess σ.

In every guess there are the two vertices $\overline{a}_j$ and $\underline{a}_j$ permanently assigned to X_j for every j and their neighborhoods $N(\overline{a}_j)$ and $N(\underline{a}_j)$ are excluded from X_j. Now it follows from properties (i) and (ii) of Lemma 10.4.3 that for each vertex $a' \in A_L$ with $a' < \overline{a}_j$ (resp. $a' \in A_R$ with $a' > \underline{a}_j$) the neighborhood $N(a')$ is a subset of $N(\overline{a}_j)$ (resp. $N(\underline{a}_j)$). Thus, these vertices a' could also be inserted in X_j without any violation of the conflict structure. Therefore, we can start from the set Π_σ of profit profiles computed for (G', p^σ) and consider iteratively (in arbitrary order) the addition of a vertex $a' \in A_L$ to one of the color classes X_j, as it is usually done in dynamic programming. Each a' is considered as an addition to every profit profile $(q_1, \ldots, q_k) \in \Pi_\sigma$ and for every index j with $a' < \overline{a}_j$ yielding new profit profiles $(q_1, \ldots, q_{j-1}, q_j + p_j(a'), q_{j+1}, \ldots, q_k)$ to be added to Π_σ. An analogous procedure is performed for all vertices $a' \in A_R$ where the addition is restricted to indices j with $a' > \underline{a}_j$.

For every guess σ, the running time is dominated by the effort of computing the $\mathcal{O}((Q+1)^k)$ profit profiles of (G', p^σ) according to Lemma 10.4.1, since adding any of the $\mathcal{O}(n)$ vertices a' requires only k operations for each profit profile.

In this way, we construct the set Π_σ of all profit profiles of partial k-colorings of G for each guess σ. It remains to identify the optimal solution in the set $\Pi := \bigcup_\sigma \Pi_\sigma$ similarly as in the proof of Theorem 10.4.2. Going over all $\mathcal{O}(n^{2k})$ guesses σ, the total running time can be given from Lemma 10.4.1 as $\mathcal{O}(n^{3k+2}(Q+1)^k)$. □

For disconnected conflict graphs, we can easily paste together the profit profiles of all connected components. Note that this construction applies to general graphs.

Lemma 10.4.5. *Given a conflict graph G consisting of $c > 1$ connected components G_ℓ, $\ell = 1, \ldots, c$, each of them with a set of profit profiles Π_ℓ, where the size of each Π_ℓ is of order $\mathcal{O}((Q+1)^k)$ with $Q = \max_{1 \le j \le k} p_j(V(G))$,* Fair k-Division Under Conflicts *can be solved for G in time $\mathcal{O}((c-1)(Q+1)^{2k})$.*

Proof. We maintain a set of profit profiles Π, initialized by $\Pi := \Pi_1$, and iteratively merge each of the profit profiles $\Pi_2, \ldots, \Pi_m$ with Π. To merge a set of profit profiles Π_ℓ, we consider every pair of profiles from Π and Π_ℓ and perform a vector addition to obtain a (possibly) new profit profile which is added to Π. At most $(Q+1)^{2k}$ such pairs may exist. In each of the $c-1$ iterations the number of different profit profiles in Π remains bounded by the trivial upper bound $(Q+1)^k$. Finally, the best objective function value is determined by evaluating all profit profiles. The total running time of this procedure is of order $\mathcal{O}((c-1)(Q+1)^{2k})$. □

Running Algorithm 1 for all c components of a graph with n vertices can be done in time $\mathcal{O}(n^{3k+2}(Q+1)^k)$. Applying Lemma 10.4.5 on the resulting profit profiles, we obtain the following corollary. Note that the computational complexity does not depend on the size of the components.

Corollary 10.4.6. *For every $k \ge 1$,* Fair k-Division Under Conflicts *is solvable in time $\mathcal{O}(n^{3k+2}(Q+1)^k + (c-1)(Q+1)^{2k})$ for biconvex bipartite conflict graphs G consisting of c connected components, where $Q = \max_{1 \le j \le k} p_j(V(G))$.*

Note that the increased running time factor of $(Q+1)^{2k}$ cannot be easily avoided. In particular, the natural idea of connecting the biconvex components by inserting dummy vertices to obtain a single connected biconvex graph does not work, as we show in the rest of this section. Note first that biconvex bipartite graphs were characterized by forbidden induced subgraphs by Tucker in [229] and the list of forbidden induced subgraphs includes all cycles except the cycle of length four and five additional graphs, including the two graphs F_1 and F_2 depicted in Fig. 10.2.

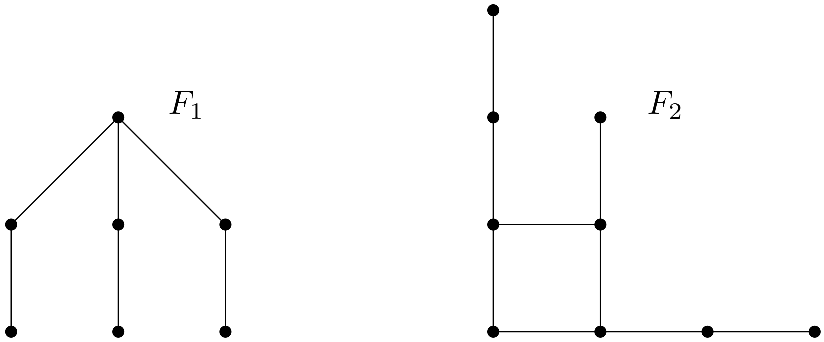


Figure 10.2: Two forbidden induced subgraphs for biconvex bipartite graphs.

Proposition 10.4.7. *There exists a disconnected biconvex bipartite graph that is not an induced subgraph of any connected biconvex bipartite graph.*

Proof. Consider the graph G depicted in Fig. 10.3.

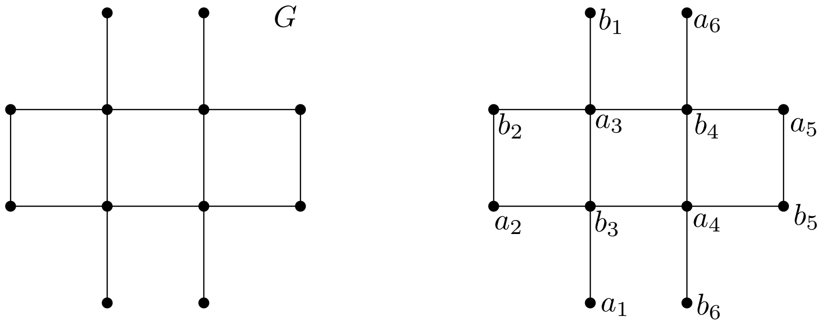


Figure 10.3: A 12-vertex biconvex bipartite graph and a biconvex labeling of it.

As shown by the vertex labeling in the figure, G is a biconvex bipartite graph. Consequently, the graph $G+K_2$, the disjoint union of G and a complete graph of order two, is also a biconvex bipartite graph. We will show that $G+K_2$ is not an induced subgraph of any connected biconvex bipartite graph.

Fix a labeling of G as in Fig. 10.3, take a disjoint copy of K_2, call it G', and suppose for a contradiction that the disjoint union $G+G'$ is an induced subgraph of a connected biconvex bipartite graph H. Let A and B denote the two parts of a bipartition of H so that $\{a_1,\ldots,a_6\} \subseteq A$ (and then $\{b_1,\ldots,b_6\} \subseteq B$).

Since H is connected, it contains a path from $V(G')$ to $V(G)$. Let P be a shortest such path. Since the sets $V(G)$ and $V(G')$ are disjoint and the are no edges between them, P has at least three vertices. Let x be the only vertex on P that has a neighbor in G, let y be the neighbor of x on P such that $y \notin V(G)$, and let z be defined as follows:

$$z = \begin{cases} \text{the neighbor of } y \text{ on } P \text{ other than } x, & \text{if } P \text{ has at least 4 vertices;} \\ \text{the neighbor of } y \text{ in } G', & \text{if } P \text{ has exactly three vertices.} \end{cases}$$

Since H is bipartite, it contains no cycle of length three. This implies that vertices x and z are not adjacent to each other.

By symmetry of G, we may assume that $x \in A$ (and thus $y \in B$ and $z \in A$). Furthermore, by the minimality of P, vertices y and z do not have any neighbors in $V(G)$. We make a series of observations about the neighborhood of x in $V(G)$.

- Vertex x cannot be adjacent to both b_3 and b_4, since otherwise H would contain an induced F_1 with vertex set $\{x, y, z, b_3, a_2, b_4, a_5\}$.

  By symmetry, we may assume that x is not adjacent to b_4.

- Vertex x is not adjacent to b_5. Suppose that it is. Then x is not adjacent to b_3, since otherwise the set $\{x, b_3, a_3, b_4, a_5, b_5\}$ would induce a 6-cycle in H. But now, H contains an induced F_1 with vertex set $\{x, b_5, a_4, b_3, a_2, b_4, a_6\}$, a contradiction.

- Vertex x is adjacent to b_3. Suppose that this is not the case. Then x is not adjacent to b_i for $i \in \{1, 2\}$, since otherwise H would contain an induced F_1 with vertex set $\{x, b_i, a_3, b_3, a_1, b_4, a_5\}$. Therefore, the only possible neighbor of x in $V(G)$ is b_6. But now, H contains an induced F_1 with vertex set $\{x, b_6, a_4, b_3, a_1, b_4, a_5\}$, a contradiction.

- Vertex x is adjacent to b_2, since otherwise H would contain an induced F_1 with vertex set $\{y, x, b_3, a_2, b_2, a_4, b_5\}$.

To conclude the proof, we observe that H contains an induced F_2 with vertex set $\{z, y, x, b_2, a_3, b_3, a_1, b_4, a_5\}$, a contradiction. □

10.4.3 Chordal graphs

In this section we present a pseudo-polynomial time algorithm that solves the FAIR k-DIVISION UNDER CONFLICTS on chordal graphs. Recall that a graph is *chordal* if all its induced cycles are of length three. First we recall some definitions and state some known results on chordal graphs and their tree decompositions.

Recall that a *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ where T is a tree whose every node t is assigned a vertex subset $X_t \subseteq V(G)$ called a bag such that the following conditions are satisfied:

- Every vertex of G is in at least one bag.

- For every edge $\{u, v\} \in E(G)$ there exists a node $t \in V(T)$ such that X_t contains both u and v.

- For every vertex $u \in V(G)$ the subgraph of T induced by the set $\{t \in V(T) : u \in X_t\}$ is connected (that is, a tree).

A tree decomposition $(T, \{X_t\}_{t \in V(T)})$ is *rooted* if we distinguish one vertex r of T which will be the root of T. This introduces natural parent-child and ancestor-descendant relations in the tree T. Following [82], we will say that a tree decomposition $(T, \{X_t\}_{t \in V(T)})$ is *nice* if it is rooted and the following conditions are satisfied:

- If $t \in V(T)$ is the root or a leaf of T, then $X_t = \emptyset$;

- Every non-leaf node t of T is one of the following three types:

- **Introduce node:** a node t with exactly one child t' such that $X_t = X_{t'} \cup \{v\}$ for some vertex $v \in V(G) \setminus X_{t'}$;
- **Forget node:** a node t with exactly one child t' such that $X_t = X_{t'} \setminus \{v\}$ for some vertex $v \in X_{t'}$;
- **Join node:** a node t with exactly two children t_1 and t_2 such that $X_t = X_{t_1} = X_{t_2}$.

The *width* of a tree decomposition $(T, \{X_t\}_{t\in V(T)})$ of a graph G is defined as $\max_{t\in V(T)} |X_t| - 1$. Lemma 7.4 from [82] shows that every tree decomposition of width at most ℓ can be transformed in polynomial time into a nice tree decomposition of width at most ℓ. The proof actually shows the following statement, which will be useful for our purpose.

Lemma 10.4.8. *Given a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t\in V(T)})$ of an n-vertex graph G, one can in time $\mathcal{O}(n^2 \cdot \max\{n, |V(T)|\})$ compute a nice tree decomposition $\mathcal{T}'$ of G that has at most $\mathcal{O}(n^2)$ nodes and such that every bag of $\mathcal{T}'$ is a subset of a bag of $\mathcal{T}$.*

Let us now apply these concepts to chordal graphs. A *clique tree* of a graph G is a tree decomposition $(T, \{X_t\}_{t\in V(T)})$ such that the bags are exactly the maximal cliques of G. It is well known (see, e.g., [27]) that a graph is chordal if and only if it has a clique tree, and in such a case a clique tree can be constructed in linear time (see, e.g., [220]). Furthermore, every chordal graph G has at most $|V(G)|$ maximal cliques (see, e.g., [27]).

Lemma 10.4.9. *Given an n-vertex chordal graph G, we can compute in linear time a tree decomposition $(T, \{X_t\}_{t\in V(T)})$ of G with $\mathcal{O}(n)$ bags, all of which are cliques.*

Combining Lemmas 10.4.8 and 10.4.9 yields the following.

Lemma 10.4.10. *Given an n-vertex chordal graph G, we can compute in time $\mathcal{O}(n^3)$ a nice tree decomposition $(T, \{X_t\}_{t\in V(T)})$ of G with $\mathcal{O}(n^2)$ bags, all of which are cliques.*

We will also need the following technical lemma about tree decompositions (see, e.g., [82]).

Lemma 10.4.11. *Let $(T, \{X_t\}_{t\in V(T)})$ be a tree decomposition of a graph G and let $\{a, b\}$ be an edge of T. The forest $T - \{a, b\}$ obtained from T by deleting edge $\{a, b\}$ consists of two connected components T_a (containing a) and T_b (containing b). Let $A = \left(\bigcup_{t\in V(T_a)} X_t\right) \setminus (X_a \cap X_b)$ and $B = \left(\bigcup_{t\in V(T_b)} X_t\right) \setminus (X_a \cap X_b)$. Then no vertex in A is adjacent to a vertex in B.*

Before we proceed to the main result for chordal graphs, we need to introduce an auxiliary definition. Let $G = (V, E)$ be a graph, let $U \subseteq V$, let $c = (X_1, \ldots, X_k)$ be a partial k-coloring of $G[X]$, and let $c' = (Y_1, \ldots, Y_k)$ be a partial k-coloring of G. We say that c' *agrees with* c *on* U if $X_j \cap U = Y_j$ for all $j \in \{1, \ldots, k\}$.

Theorem 10.4.12. *For every $k \geq 1$,* FAIR k-DIVISION UNDER CONFLICTS *is solvable in time $\mathcal{O}(n^{k+2}(Q + 1)^{2k})$ for a chordal conflict graph G, where $Q = \max_{1\leq j\leq k} p_j(V(G))$.*

Proof. Fix $k \geq 1$ and let G be a chordal graph equipped with profit functions $p_1, \ldots, p_k : V(G) \to \mathbb{Z}_+$. We will show that we can compute the set Π of all profit profiles of partial k-colorings of G in the stated running time. The maximum satisfaction level over all profit profiles will then give the optimal value of FAIR k-DIVISION UNDER CONFLICTS for $(G, p_1, \ldots, p_k)$.

We first apply Lemma 10.4.10 and compute in time $\mathcal{O}(n^3)$ a nice tree decomposition $(T, \{X_t\}_{t \in V(T)})$ of G with $\mathcal{O}(n^2)$ bags, all of which are cliques. Recall that by definition T is a rooted tree decomposition of G. Let r be the root of T. For every node $t \in V(T)$, we denote by V_t the union of all bags $X_{t'}$ such that $t' \in V(T)$ is a (not necessarily proper) descendant of t in T.

We traverse tree T bottom-up and use a dynamic programming approach to compute, for every node $t \in V(T)$ and every partial k-coloring c of $G[X_t]$, the family $P(t, c)$ of all profit profiles of partial k-colorings of $G[V_t]$ that agree with c on X_t.

Since $(T, \{X_t\}_{t \in V(T)})$ is a nice tree decomposition, we have $X_r = \emptyset$; in particular, the *trivial* partial k-coloring $\emptyset^k$ consisting of k empty sets is the only partial k-coloring of $G[X_r]$. Thus, since $V_r = V(G)$ and every partial k-coloring of G agrees with the trivial partial k-coloring of $G[X_r]$ on X_r, the set $P(r, \emptyset^k)$ is the set of all profit profiles of partial k-colorings of G, which is what we want to compute.

We consider various cases depending on the type of a node $t \in V(T)$ in the nice tree decomposition. For each of them we give a formula for computing the set $P(t, c)$ from the already computed sets of the form $P(t', c')$ where t' is a child of t in T, and argue why the formula is correct.

1. **t is a leaf node.**

   By the definition of a nice tree decomposition it follows that $X_t = \emptyset$. Thus, the only partial k-coloring of $G[X_t]$ is the trivial one, $\emptyset^k$. Clearly, $P(t, \emptyset^k) = \{(0, \ldots, 0)\}$.

2. **t is an introduce node.**

   By definition, t has exactly one child t' and $X_t = X_{t'} \cup \{v\}$ holds for some vertex $v \in V \setminus X_{t'}$. Clearly, $V_t = V_{t'} \cup \{v\}$, and this is a disjoint union. (If $v \in V_{t'}$, then the subtree of T consisting of all bags X_τ such that $v \in X_\tau$ is not connected; a contradiction.) Consider an arbitrary partial k-coloring $c = (X_1, \ldots, X_k)$ of $G[X_t]$. We want to compute $P(t, c)$ using the set $P(t', c')$, where $c' = (X_1 \setminus \{v\}, \ldots, X_k \setminus \{v\})$. (Note that c' is a partial k-coloring of $G[X_{t'}]$.) We claim that the following equality holds:

   $$P(t, c) = \begin{cases} \{\mathbf{q} + \mathbf{e}_j(p_j(v)) \mid \mathbf{q} \in P(t', c')\}, & \text{if } v \in X_j \text{ for some } j \in \{1, \ldots, k\}; \\ P(t', c'), & \text{otherwise.} \end{cases}$$

   To show the recurrence, note first that if for all $j \in \{1, \ldots, k\}$ we have $v \notin X_j$, then $c' = c$ and thus $P(t, c) = P(t', c')$ in this case. If, however, $v \in X_j$ for some $j \in \{1, \ldots, k\}$, then there can only be one such j, and thus $c' = (X_1, \ldots, X_{j-1}, X_j \setminus \{v\}, X_{j+1}, \ldots, X_k)$. In this case, we will need the fact that v is not adjacent to any vertex of $V_{t'} \setminus X_{t'}$. Indeed, applying Lemma 10.4.11 to $a = t$ and $b = t'$ shows that no vertex of $V(G) \setminus V_{t'}$ is adjacent to any vertex of $V_{t'} \setminus X_{t'}$, hence the statement follows since $v \in V(G) \setminus V_{t'}$.

The fact that all neighbors of v in the set $V_{t'}$ are contained in $X_{t'}$ implies that for every partial k-coloring of $G[V_{t'}]$ that agrees with c' on $X_{t'}$, adding v to the j-th color class will result in a partial k-coloring of $G[V_t]$ that agrees with c on X_t. Thus, there is a bijective correspondence between the set of partial k-colorings of $G[V_t]$ that agree with c on X_t and those of $G[V_{t'}]$ that agree with c' on $X_{t'}$, given by removing v from the j-th color class. This implies the claimed equality $P(t,c) = \{\mathbf{q} + \mathbf{e}_j(p_j(v)) \mid \mathbf{q} \in P(t',c')\}$.

3. **t is a forget node.**

   By definition, t has exactly one child t' in T and $X_t = X_{t'} \setminus \{v\}$ holds for some vertex $v \in V \setminus X_t$. Thus, $V_t = V_{t'}$. Consider an arbitrary partial k-coloring $c = (X_1, \ldots, X_k)$ of $G[X_t]$. We claim that the following equality holds:

   $$P(t,c) = P(t',c) \cup \bigcup_{j : X_j = \emptyset} P(t', (X_1, \ldots, X_{j-1}, \{v\}, X_{j+1} \ldots, X_k)) .$$

   Consider an arbitrary partial k-coloring $(Y_1, \ldots, Y_k)$ of $G[V_t]$ that agrees with c on X_t. If $v \notin Y_j$ for all $j \in \{1, \ldots, k\}$, then $(Y_1, \ldots, Y_k)$ agrees with c on $X_{t'}$. Suppose now that $v \in Y_j$ for some $j \in \{1, \ldots, k\}$. Then, j is unique. Furthermore, since $X_{t'}$ is a clique in G and hence in $G[V_{t'}]$, the fact that $v \in Y_j$ implies that $Y_j \cap X_{t'} = \{v\}$, and consequently $X_j = Y_j \cap X_t = \emptyset$. In this case, the partial k-coloring $(Y_1, \ldots, Y_k)$ agrees with the partial k-coloring $(X_1, \ldots, X_{j-1}, \{v\}, X_{j+1}, \ldots, X_k)$ of $G[V_{t'}]$ on $X_{t'}$. Thus, every partial k-coloring of $G[V_t]$ that agrees with c on X_t either agrees with c on $X_{t'}$ or agrees with $(X_1, \ldots, X_{j-1}, \{v\}, X_{j+1} \ldots, X_k)$ on $X_{t'}$ for some $j \in \{1, \ldots, k\}$ such that $X_j = \emptyset$. Similar arguments can be used to show the converse inclusion, that is, any partial k-coloring of $G[V_{t'}]$ that satisfies one of the above conditions is a partial k-coloring of $G[V_t]$ that agrees with c on X_t. This implies the claimed equality.

4. **t is a join node.**

   By definition, t has exactly two children t_1 and t_2 in T and it holds that $X_t = X_{t_1} = X_{t_2}$. We claim that $V_{t_1} \cap V_{t_2} = X_t$. It is clear that $X_t \subseteq V_{t_1} \cap V_{t_2}$. Assume for contradiction that there is a vertex $v \in V(G)$ such that $v \in (V_{t_1} \cap V_{t_2}) \setminus X_t$. Then there are nodes t_1' and t_2' of T such that $v \in X_{t_1'}$, $v \in X_{t_2'}$, and t_1' and t_2' are (possibly not proper) descendants of t_1 and t_2, respectively. It follows that the subgraph of T consisting of all bags containing v is not connected; a contradiction. Thus $X_t = V_{t_1} \cap V_{t_2}$, as claimed. Furthermore, applying Lemma 10.4.11 to $a = t_1$ and $b = t$ we can show that no vertex of $V_{t_1} \setminus X_t$ is adjacent in G to any vertex of $V(G) \setminus V_{t_1}$. Since $V_{t_2} \setminus X_t \subseteq V(G) \setminus V_{t_1}$, this implies that no vertex in $V_{t_1} \setminus X_t$ is adjacent in G to any vertex of $V_{t_2} \setminus X_t$.

   Consider now an arbitrary partial k-coloring $c = (X_1, \ldots, X_k)$ of $G[X_t]$ (observe that c is also a partial k-coloring of $G[X_{t_1}]$ and $G[X_{t_2}]$). In this case, we have the following recurrence relation:

   $$P(t,c) = \{\mathbf{q_1} + \mathbf{q_2} - (p_1(X_1), \ldots, p_k(X_k)) \mid \mathbf{q_1} \in P(t_1,c), \mathbf{q_2} \in P(t_2,c)\} .$$

   It is clear that for any partial k-coloring $(X_1', \ldots, X_k')$ of $G[V_t]$ that agrees with c on X_t, the k-tuples $(X_1' \cap V_{t_1}, \ldots, X_k' \cap V_{t_1})$ and $(X_1' \cap V_{t_2}, \ldots, X_k' \cap V_{t_2})$ are partial k-colorings of $G[V_{t_1}]$ and $G[V_{t_2}]$ that agree with c on X_{t_1} and X_{t_2}, respectively.

The fact that no vertex in $V_{t_1} \setminus X_t$ is adjacent in G to any vertex in $V_{t_2} \setminus X_t$ implies that the other direction is also true: given partial k-colorings $(X'_1, \ldots, X'_k)$ and $(X''_1, \ldots, X''_k)$ of $G[V_{t_1}]$ and $G[V_{t_2}]$ that agree with c on X_{t_1} and X_{t_2}, respectively, we have $X'_j \cap X_t = X''_j \cap X_t = X_j$ for all $j \in \{1, \ldots, k\}$, and thus $(X'_1 \cup X''_1, \ldots, X'_k \cup X''_k)$ is a partial k-coloring of $G[V_t]$ that agrees with c on X_t. Furthermore, for all $j \in \{1, \ldots, k\}$, the fact that $V_{t_1} \cap V_{t_2} = X_t$ implies that $X'_j \cap X''_j = X_j$, and hence $p_j(X'_j \cup X''_j) = p_j(X'_j) + p_j(X''_j) - p_j(X_j)$. The claimed equality follows.

It remains to estimate the time complexity of the algorithm. We compute a nice tree decomposition of G in time $\mathcal{O}(n^3)$. Each of the $\mathcal{O}(n^2)$ bags is a clique, so in total we have $\mathcal{O}(n^k)$ partial k-colorings per bag. Furthermore, note that for each partial coloring $(X_1, \ldots, X_k)$ of any induced subgraph of G and each $j \in \{1, \ldots, k\}$, we have $p_j(X_j) \in \{0, 1, \ldots, Q\}$. Thus, each set $P(t, c)$ has at most $(Q+1)^k$ elements. For each of the $\mathcal{O}(n^{k+2})$ pairs (t, c) where t is a node of T and c is a partial k-coloring of $G[X_t]$, we compute the set $P(t, c)$ using the formula corresponding to the type of node t. The time complexity of this step depends on the type of the node. Case 1 takes constant time. In Case 2, we check in constant time whether $v \in X_j$ for some $j \in \{1, \ldots, k\}$ and then compute the set $P(t, c)$ in time $\mathcal{O}((Q+1)^k)$. In Case 3, we first compute in (constant) time $\mathcal{O}(k)$ the set of indices $j \in \{1, \ldots, k\}$ such that $X_j = \emptyset$. Then, the union given by the formula can be computed in time $\mathcal{O}((Q+1)^k)$, simply by iterating over all families in the union and keeping track of which of the $\mathcal{O}((Q+1)^k)$ profit profiles appear in any of the families. Finally, Case 4 can be done in time $\mathcal{O}((Q+1)^{2k})$. Altogether, this results in running time $\mathcal{O}((Q+1)^{2k})$ for each fixed $t \in V(T)$ and each partial k-coloring c of X_t. Consequently, the total running time of the algorithm is $\mathcal{O}(n^{k+2}(Q+1)^{2k})$. □

10.4.4 Graphs with bounded treewidth

Recall that the width of a tree decomposition $(T, \{X_t\}_{t \in V(T)})$ of a graph G is defined as $\max_{t \in V(T)} |X_t| - 1$. The *treewidth* of a graph G is the minimum possible width of a tree decomposition of G. A graph class $\mathcal{G}$ is said to be of *bounded treewidth* if there exists a nonnegative integer ℓ such that each graph in $\mathcal{G}$ has treewidth at most ℓ. For each fixed treewidth bound ℓ, given a graph G of treewidth at most ℓ, a tree decomposition of G of width at most ℓ can be computed in linear time [31]. Such a decomposition leads to linear-time algorithms for many problems that are generally NP-hard (see, e.g., [11, 77]).

A similar approach as the one used in the proof of Theorem 10.4.12 for solving the Fair k-Division Under Conflicts on chordal graphs can be used on graphs of bounded treewidth.

Fix $k, \ell \geq 1$ and let $(G, p_1, \ldots, p_k)$ be the input to Fair k-Division Under Conflicts such that the treewidth of G is at most ℓ. In time $\ell^{\mathcal{O}(\ell^3)} n$ we can compute a tree decomposition of G a width at most ℓ using the algorithm of Bodlaender [31]. Clearly, the obtained tree decomposition has at most $\ell^{\mathcal{O}(\ell^3)} n$ bags. By Lemma 10.4.8 it follows that we can compute in time $\mathcal{O}(\ell^{\mathcal{O}(\ell^3)} n^3)$ a nice tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of G of width at most ℓ, with $\mathcal{O}(n^2)$ bags. Every bag has at most $\ell + 1$ vertices, so for every bag we have at most a constant number, $(\ell+1)^{k+1}$, partial k-colorings, which in total gives $\mathcal{O}(n^2)$ pairs (t, c) of a node $t \in V(T)$ and a partial k-coloring c of t. For each such pair (t, c), we again compute the family $P(t, c)$ of all profit profiles of partial k-colorings of $G[V_t]$ that agree with c on X_t. Since $\mathcal{T}$ is a nice tree decomposition, every node is of one of the four possible types, and in Cases 1, 2, and 4 we have identical equalities as in the corresponding cases in the proof of Theorem 10.4.12, while in Case 3 the union over all j such that $X_j = \emptyset$ of the sets $P(t', (X_1, \ldots, X_{j-1}, \{v\}, X_{j+1} \ldots, X_k))$ is replaced by the union over all j such that $X_j \cup \{v\}$ is an independent set in G of the sets $P(t', (X_1, \ldots, X_{j-1}, X_j \cup \{v\}, X_{j+1} \ldots, X_k))$. Since we can compute the adjacency matrix of G in time $\mathcal{O}(n^2)$, we may assume that adjacency checks can be done in constant time. Thus, the expressions in the formulas corresponding to each of the Cases 2 and 3 can be evaluated in time $\mathcal{O}((Q+1)^k)$, while the corresponding time complexity of Case 4 is $\mathcal{O}((Q+1)^{2k})$. Altogether, this gives us the claimed running time and yields the following theorem (where the constant hidden in the $\mathcal{O}$ notation depends on k and ℓ).

Theorem 10.4.13. *For every $k \geq 1$ and $\ell \geq 1$,* Fair k-Division Under Conflicts *is solvable in time $\mathcal{O}(n^2(n + (Q+1)^{2k}))$ for a graph G of treewidth at most ℓ, where $Q = \max_{1 \leq j \leq k} p_j(V(G))$.*

It turns out that this result can be generalized by constructing the polynomial-time algorithm for graphs of bounded clique-width.

10.4.5 Graphs of bounded clique-width

In this section we present a pseudo-polynomial time dynamic programming algorithm for Fair k-Division Under Conflicts for conflict graphs of *bounded clique-width.* This is an improvement over the result for graphs of bounded treewidth, which was so far the only positive result for non-perfect graphs.

Clique-width, introduced in 1993 [79], is a parameter defined by a construction process where only a limited number of vertex labels are available. Vertices with the same label at some point must be treated uniformly in subsequent steps (see below). The clique-width $\mathrm{cw}(G)$ of a graph G is the minimum number of labels that suffice to construct G in this way. NP-completeness and inapproximability of the clique-width of a graph were shown in [105]. For graphs of bounded clique-width many hard optimization problems admit polynomial-time algorithms, see, e.g., [80, 99, 124, 210].

Relations between treewidth and clique-width were elaborated in [81]. In particular, bounded treewidth $\mathrm{tw}(G)$ of a graph G implies bounded clique-width since $\mathrm{cw}(G) \leq 3 \cdot 2^{\mathrm{tw}(G)} - 1$ as shown by [74]. However, the opposite implication is not true as can be seen from the family of complete graphs which have clique-width 2 but treewidth $|V| - 1$.

Another parameter of a graph G related to treewidth is *rank-width* $\mathrm{rw}(G)$ introduced in [191]. Rank-width is also derived from a hierarchical decomposition of the graph. Informally speaking, treewidth measures the width of a separation into two sides, whereas rank-width measures the rank of the adjacency matrix of the edges between the two sides of the separation. Without going into more details, let us just mention that it was shown in [191] that $\mathrm{rw}(G) \leq \mathrm{cw}(G) \leq 2^{\mathrm{rw}(G)+1} - 1$. Therefore, bounded clique-width is equivalent to bounded rank-width.

In the following we will describe the labelling process of the graph decomposition associated to clique-width in more detail.

A *labeled graph* is a graph in which every vertex is assigned some label from $\mathbb{N}$. If all vertex labels belong to the set $[k]$, then we say that the graph is k-labeled. The *clique-width* of a graph G is defined as the smallest positive integer k such that a k-labeled graph isomorphic to G can be constructed with the following operations:

- $i(v)$: creating a new one-vertex graph with vertex v labeled i,
- $G \oplus H$: disjoint union of two already constructed labeled graphs G and H,
- $\eta_{i,j}$, for $i \neq j$: adding to G all edges between vertices labeled i and vertices labeled j,
- $\rho_{i \to j}$, for $i \neq j$: relabeling every vertex labeled i with label j.

A construction of a graph G with the above four operations can be represented by an algebraic expression, which is called a k*-expression* if it uses at most k labels. Given a k-expression σ, we denote by $|\sigma|$ its encoding length. A graph class $\mathcal{G}$ is said to be of *bounded clique-width* if there exists a nonnegative integer k such that each graph in $\mathcal{G}$ has clique-width at most k.

Polynomial-time algorithms for graphs with bounded clique-width are typically developed using dynamic programming based on a k-expression building the input graph. If a k-expression is not available, then one can use any of the available algorithms in the literature for computing an expression with at most $f(k)$ labels for some exponential function f (see [109, 145, 190, 191]). The currently fastest such algorithm is due to Fomin and Korhonen [109]; for an integer k and an n-vertex graph G, it runs in time $2^{2^{O(k)}}n^2$ and either computes a $(2^{2k+1}-1)$-expression of G or correctly determines that the clique-width of G is more than k.

We can now proceed to prove the following theorem.

Theorem 10.4.14. *For every two positive integers k and ℓ,* Fair k-Division Under Conflicts *is solvable in time $\mathcal{O}\left(4^{k\ell}|\sigma|(Q+1)^{2k}\right)$, if the conflict graph G has clique-width at most ℓ and is given by an ℓ-expression σ, where $Q = \max_{1\leq j\leq k} p_j(V(G))$.*

Proof. We extend the standard dynamic programming algorithm for graphs of bounded clique-width for the case $k = 1$, that is, the maximum weight independent set problem (see, e.g., [131]). Given a partial k-coloring $c = (X_1, \ldots, X_k)$ of an ℓ-labeled graph H, the *label profile* of c (with respect to H) is the k-tuple $(L_1, \ldots, L_k)$ where L_j is the set of labels in $[\ell]$ appearing on some vertex of X_j, for all $j \in [k]$. For each labeled subgraph H of G that appears in the process of constructing G using σ and each k label sets $L_1, \ldots, L_k \subseteq [\ell]$, we compute the set $P(H, L_1, \ldots, L_k)$ of all profit profiles $(q_1, \ldots, q_k)$ of partial k-colorings c of H such that the label profile of c equals $(L_1, \ldots, L_k)$. We then have four cases depending on the type of H. In each case, we derive a formula of how to compute the set $P(H, L_1, \ldots, L_k)$ from the previously computed sets of this type.

1. **H is a one-vertex graph consisting of a vertex v labeled i.**

   There are only $k+1$ partial k-colorings of H: the trivial partial k-coloring $\emptyset^k$ consisting of k empty sets, and, for each $j \in [k]$, the partial k-coloring $c_j = (X_1, \ldots, X_k)$ where $X_j = \{v\}$ and $X_{j'} = \emptyset$ for all $j' \in [k] \setminus \{j\}$. The label profile of $\emptyset^k$ is $\emptyset^k$. For each $j \in [k]$, the label profile of c_j is the k-tuple $(L_1, \ldots, L_k)$ where $L_j = \{i\}$ and $L_{j'} = \emptyset$ for all $j' \neq j$. Thus, denoting by $\mathbf{e}_j(p_j(v))$ the k-tuple in $\mathbb{Z}_+^k$ with j-th coordinate equal to $p_j(v)$ and all the other coordinates equal to 0, we have the following formula:

$$P(H, L_1, \ldots, L_k) = \begin{cases} \{\mathbf{e}_j(p_j(v))\}, & \text{if } L_j = \{i\} \text{ and } L_{j'} = \emptyset \text{ for all } j' \neq j\,, \\ \{(0, \ldots, 0)\}, & \text{if } L_j = \emptyset \text{ for all } j \in [k]\,, \\ \emptyset, & \text{otherwise.} \end{cases}$$

While in the remaining three cases, the assumptions on H are different, we always describe how to compute the set $P(H, L_1, \ldots, L_k)$ for an arbitrary but fixed collection of k label sets $L_1, \ldots, L_k \subseteq [\ell]$.

2. **H is the disjoint union of two labeled graphs H_1 and H_2.**

   Let $c = (X_1, \ldots, X_k)$ be a partial k-coloring of H with label profile $(L_1, \ldots, L_k)$. Then for $i \in \{1, 2\}$ we have that $c_i = (X_1 \cap V(H_i), \ldots, X_k \cap V(H_i))$ is a partial k-coloring of H_i. Let us denote by $(L'_1, \ldots, L'_k)$ and $(L''_1, \ldots, L''_k)$ the label profiles of c_1 and c_2, respectively. Then $L_j = L'_j \cup L''_j$ for all $j \in [k]$. Furthermore, the converse direction holds as well: for any two partial k-colorings $c_1 = (X'_1, \ldots, X'_k)$ and $c_2 = (X''_1, \ldots, X''_k)$ of H_1 and H_2, respectively, the k-tuple $c = (X'_1 \cup X''_1, \ldots, X'_k \cup X''_k)$ is a partial k-coloring of H with label profile $(L'_1 \cup L''_1, \ldots, L'_k \cup L''_k)$, where $(L'_1, \ldots, L'_k)$ and $(L''_1, \ldots, L''_k)$ are the label profiles of c_1 and c_2, respectively. This bijective correspondence yields the following formula:

   $$\begin{aligned} &P(H, L_1, \ldots, L_k) = \\ &= \bigcup \{\mathbf{q_1} + \mathbf{q_2} \mid \mathbf{q_1} \in P(H_1, L'_1, \ldots, L'_k), \mathbf{q_2} \in P(H_2, L''_1, \ldots, L''_k)\}\,, \end{aligned}$$

   where the union is taken over all collections $(L'_1, \ldots, L'_k)$ and $(L''_1, \ldots, L''_k)$ of label sets such that $L'_j \cup L''_j = L_j$ for all $j \in [k]$.

3. **H is obtained from a labeled graph H' by adding all edges between vertices labeled i and vertices labeled j where $i \neq j$.**

   Assume first that there exists some $s \in [k]$ such that $\{i, j\} \subseteq L_s$ and let $c = (X_1, \ldots, X_k)$ be a partial k-coloring of H with label profile $(L_1, \ldots, L_k)$. Since $\{i, j\} \subseteq L_s$, there are vertices v_1 and v_2 of H labeled i and j, respectively, such that $\{v_1, v_2\} \subseteq X_s$. By the assumption on H all vertices labeled i are adjacent in H to all vertices labeled j, so it is not possible that $\{v_1, v_2\} \subseteq X_s$, since X_s is an independent set in H; a contradiction. It follows that there is no partial k-coloring of H with label profile $(L_1, \ldots, L_k)$, so in this case $P(H, L_1, \ldots, L_k) = \emptyset$.

   Assume now that for every $s \in [k]$ we have that $|L_s \cap \{i, j\}| \leq 1$. In this case, every partial k-coloring of H with label profile $(L_1, \ldots, L_k)$ is also a partial k-coloring of H' with the same label profile (with respect to H'), and vice versa. It follows that $P(H, L_1, \ldots, L_k) = P(H', L_1, \ldots, L_k)$.

   Altogether, we have the following equality:

   $$P(H, L_1, \ldots, L_k) = \begin{cases} P(H', L_1, \ldots, L_k), & \text{if } |\{i, j\} \cap L_s| \leq 1 \text{ for all } s \in [k], \\ \emptyset, & \text{otherwise.} \end{cases}$$

4. **H is obtained from a labeled graph H' by relabeling all vertices labeled i to vertices labeled j.**

   Let $c = (X_1, \ldots, X_k)$ be a partial k-coloring of H with label profile $(L_1, \ldots, L_k)$. Observe that it follows from the assumption on H that no vertex in H has label i.

   If there exists some $s \in [k]$ such that $i \in L_s$, then there is a vertex $v \in X_s$ labeled i; a contradiction. It follows that there is no partial k-coloring of H with label profile $(L_1, \ldots, L_k)$, and we have that $P(H, L_1, \ldots, L_k) = \emptyset$ in this case.

Assume now that for all $s \in [k]$ we have that $i \notin L_s$. Let $I = \{s \in [k] \mid j \in L_s\}$ and consider an arbitrary $s \in I$. The vertices in X_s form an independent set in H and thus also in H'. Since $j \in I$, the set L_s contains j and therefore there exists a vertex $v \in X_s$ such that the label of v in H is j. Thus, the set X_{sj} of all vertices in X_s labeled j in H is nonempty. Furthermore, since the label in H' of any vertex in X_{sj} is either i or j, the label set of X_s in H' depends on whether there exists a vertex in X_{sj} labeled i in H' and whether there exists a vertex in X_{sj} labeled j in H'. More precisely, the dependency is as follows.

- If there exists a vertex in X_{sj} labeled i in H' as well as one labeled j in H', then the label set of X_s in H' is $L_s \cup \{i\}$ (recall that $j \in L_s$).
- If all vertices in X_{sj} are labeled i in H', then the label set of X_s in H' is $(L_s \setminus \{j\}) \cup \{i\}$.
- If all vertices in X_{sj} are labeled j in H', then the label set of X_s in H' is L_s.

We conclude that the vertices of X_s form in H' an independent set with label set being equal either to L_s, to $(L_s \setminus \{j\}) \cup \{i\}$, or to $L_s \cup \{i\}$. Therefore, c is a partial coloring of H' with label profile $(L'_1, \ldots, L'_k)$ such that for all $s \in I$ we have $L'_s \in \{L_s, (L_s \setminus \{j\}) \cup \{i\}, L_s \cup \{i\}\}$, and for all $s \in [k] \setminus I$ we have $L'_s = L_s$. Conversely, for any k label sets $L'_1, \ldots, L'_k \subseteq [\ell]$ such that $L'_s \in \{L_s, (L_s \setminus \{j\}) \cup \{i\}, L_s \cup \{i\}\}$ for all $s \in I$ and $L'_s = L_s$ for all $s \in [k] \setminus I$, any partial coloring of H' with label profile $(L'_1, \ldots, L'_k)$ is a partial coloring of H with label profile $(L_1, \ldots, L_k)$.

Altogether, we thus obtain the following equality:

$$P(H, L_1, \ldots, L_k) = \begin{cases} \emptyset, & \text{if } i \in L_s \text{ for some } s \in [k] \\ \bigcup P(H', L'_1, \ldots, L'_k), & \text{otherwise,} \end{cases}$$

where the union in the second case is taken over all k-tuples of label sets $L'_1, \ldots, L'_k \subseteq [\ell]$ such that for all $s \in I$ we have $L'_s \in \{L_s, (L_s \setminus \{j\}) \cup \{i\}, L_s \cup \{i\}\}$, and for all $s \in [k] \setminus I$ we have $L'_s = L_s$.

Time complexity analysis. From the ℓ-expression σ we compute in time $|\sigma|$ a rooted tree T describing the construction of G. Each node of T corresponds to a labeled subgraph H of G. For each such subgraph H we consider all the $2^{\ell k}$ different collections of k label sets $(L_1, \ldots, L_k)$, obtained by choosing a subset of $[\ell]$ for each coordinate. We explain the time complexity separately for Case 2. For Case 2, we can initialize all the sets $P(H, L_1, \ldots, L_k)$ to be empty and iterate over all $4^{k\ell}$ pairs of collections $(L'_1, \ldots, L'_k)$ and $(L''_1, \ldots, L''_k)$ of label sets of H_1 and H_2. For each such iteration we add to $P(H, L_1, \ldots, L_k)$, where $L_j = L'_j \cup L''_j$ for all $j \in [k]$, the elements of the set $\{\mathbf{q_1} + \mathbf{q_2} \mid \mathbf{q_1} \in P(H_1, L'_1, \ldots, L'_k), \mathbf{q_2} \in P(H_2, L''_1, \ldots, L''_k)\}$ in time $(Q+1)^{2k}$. Hence, the overall time complexity for a graph H in Case 2 is $\mathcal{O}(4^{k\ell}(Q+1)^{2k})$. For the remaining three cases, we estimate the running time separately for each set $P(H, L_1, \ldots, L_k)$. The expressions in the formulas for computing $P(H, L_1, \ldots, L_k)$ can be evaluated in time $\mathcal{O}(k\ell)$ in Case 1, in time $\mathcal{O}(k\ell + (Q+1)^k)$ in Case 3, and in time $\mathcal{O}(k\ell \cdot 3^k \cdot (Q+1)^k)$ in Case 4. Since Case 4 dominates the other two cases, the overall time complexity for a graph H resulting from Cases 1, 3, and 4 is given by $\mathcal{O}(2^{k\ell} \cdot k\ell \cdot 3^k (Q+1)^k)$. Since $k\ell \leq 2^{k\ell}$ and $3^k \leq (Q+1)^k$ for all $Q \geq 2$, the overall time complexity of Cases 1, 3, and 4 is dominated by the effort for Case 2, which yields the claimed running time bound.

(The special case $Q = 1$ would imply that for each agent, only one item has a non-zero profit. This could be solved trivially in time $\mathcal{O}(k)$.) □

We conclude this section with some remarks about another, more general solution approach to FAIR k-DIVISION UNDER CONFLICTS for graphs of bounded clique-width. The unweighted version of FAIR k-DIVISION UNDER CONFLICTS (in its decision version) takes a graph G and an integer q as input and asks about the existence of a partial k-coloring in G in which all the color classes have cardinality at least q. The existence of a pseudo-polynomial-time algorithm for this problem on graphs with bounded clique-width follows from a metatheorem of Courcelle and Durand [78, Theorem 27], and it is plausible that with a suitable adaptation of their approach, a solution for the general FAIR k-DIVISION UNDER CONFLICTS problem might also be developed. However, as the algorithms constructed in [78] are very general, their running times are not specified precisely. In contrast, our algorithm given in the proof of Theorem 10.4.14 is directly tailored for FAIR k-DIVISION UNDER CONFLICTS and it is not difficult to analyze its running time.

10.5 Approximation

All the pseudo-polynomial dynamic programming algorithms presented in this paper share the following characteristics. Throughout the execution feasible states are computed, where every state describes a profit allocation given by a feasible solution of FAIR k-DIVISION UNDER CONFLICTS. Each such state is represented by a k-dimensional vector $(q_1, \ldots, q_k) \in \mathbb{Z}_+^k$, where every entry q_j describes the profit $p_j(X_j)$ assigned to agent j by a partial coloring $(X_1, \ldots, X_k)$. While Pareto-dominated states can be eliminated, the total number of states remains trivially bounded by $(Q+1)^k$, where $Q = \max_{1 \le j \le k} p_j(V(G))$. The optimal solution with maximum satisfaction level can be determined at the end of such an algorithm by simply going through all generated states and inspecting their satisfaction levels.

In a canonical step of our algorithms a vertex v (resp. item) is feasibly assigned to an agent j thereby generating a new state $(q_1, \ldots, q_{j-1}, q_j + p_j(v), q_{j+1}, \ldots, q_k)$ from a previous state $(q_1, \ldots, q_k)$. The decisions taken by the algorithms depend only on the graph but not on the profit values of previously generated states. Every vertex is assigned to each agent at most once.

Under these preconditions, we can derive a fully polynomial time approximation scheme (FPTAS) for each such dynamic programming algorithm (considering k as a constant). For an optimal satisfaction level z^*, an FPTAS computes for every given $\varepsilon > 0$, an approximate solution with satisfaction level z^A fulfilling $z^A \ge z^*/(1+\varepsilon)$ with running time polynomial in the size of the encoded input and in $1/\varepsilon$.

The FPTAS is based on the observation that the k profit values of a solution can also be seen as k objective function values in a multiobjective optimization problem. Thus, the technique for deriving an FPTAS for the multiobjective knapsack problem described in [98] can be applied as follows.

Denote the upper bound for the profit assigned to agent j by $UB_j = p_j(V(G))$ and set $u_j = \lceil n log_{1+\epsilon} UB_j \rceil$, where, as usual, $n = |V(G)|$. Partition the profit range for each agent j into u_j intervals

$$[1,(1+\varepsilon)^{1/n}),\ [(1+\varepsilon)^{1/n},(1+\varepsilon)^{2/n}),[(1+\epsilon)^{2/n},(1+\varepsilon)^{3/n}),\dots \\ [(1+\varepsilon)^{(u_j-1)/n},(1+\varepsilon)^{u_j/n}]\,.$$

To obtain an FPTAS from the generic dynamic programming algorithm indicated above we restrict the possible profit values q_j allocated to agent j to the lower interval endpoints of these intervals. The FPTAS mimics exactly the operations of the exact dynamic program, but whenever a vertex v is assigned to j, the resulting profit $q_j+p_j(v)$ is *rounded down* to the nearest interval endpoint. Note that this does not change the steps of the dynamic program since we assumed that its decisions do not depend on the profit values of states.

The bound $u_j=\lceil n\log_{1+\varepsilon} UB_j\rceil$ is in $\mathcal{O}(n/\varepsilon\cdot\log_2(UB_j))$, which is polynomial in the encoding length of the input, since

$$\log_{1+\varepsilon} UB_j=(\ln 2\log_2 UB_j)/\ln(1+\varepsilon)\le(2\ln 2\log_2 UB_j)/\varepsilon\,,$$

for all $\varepsilon\in(0,1)$. The above inequality follows from $x\le 2\ln(1+x)$, which can be verified to hold for all $x\in(0,1)$ by standard calculus. Thus, the total number of states in the modified algorithm is bounded by $\mathcal{O}((n/\varepsilon)^k(\log_2 Q)^k)$.

Concerning the loss of accuracy we can proceed similarly to [98] and compare an arbitrary state $(q_1,q_2,\dots,q_k)$ of the exact dynamic program to some state of the FPTAS consisting of lower interval endpoints $(\tilde q_1,\tilde q_2,\dots,\tilde q_k)$. For every state $(q_1,\dots,q_j,\dots,q_k)$ generated by the exact algorithm after assigning i vertices to agent j, we claim that in the FPTAS there exists a state $(\tilde q_1,\tilde q_2,\dots,\tilde q_k)$ of lower interval endpoints such that

$$q_j\le(1+\varepsilon)^{i/n}\tilde q_j\,. \tag{10.5}$$

This claim can be shown by induction. For $i=1$, there was one vertex v assigned to agent j giving profit $q_j=p_j(v)$. In the FPTAS, there will be a state where $\tilde q_j$ is the largest lower interval endpoint not exceeding q_j. By construction of the intervals, we have $(1+\varepsilon)^{1/n}\tilde q_j\ge q_j$.

Assuming the claim to be true for some $i-1$, we consider the i-th assignment of a vertex v to j. In the exact algorithm, $p_j(v)$ is added to some value q_j for which there exists a lower interval endpoint $\tilde q_j$ fulfilling $q_j\le(1+\varepsilon)^{(i-1)/n}\tilde q_j$. During the FPTAS, $p_j(v)$ will also be added to $\tilde q_j$ and the result will be rounded down to a lower interval endpoint $\tilde q'$ with $(1+\varepsilon)^{1/n}\tilde q'\ge\tilde q_j+p_j(v)\ge(1+\varepsilon)^{-(i-1)/n}q_j+p_j(v)\ge(1+\varepsilon)^{-(i-1)/n}(q_j+p_j(v))$. Moving terms around, this proves (10.5) for the new profit $q_j+p_j(v)$.

Since there can be at most n vertices assigned to any agent, (10.5) holds also for the satisfaction level of the optimal solution.

Summarizing the above discussion and the proofs of Theorem 10.4.2, Corollary 10.4.6, Theorem 10.4.12, and Theorem 10.4.13, we conclude:

Theorem 10.5.1. FAIR k-DIVISION UNDER CONFLICTS *with constant k admits an FPTAS if the conflict graph is a cocomparability graph, a biconvex bipartite graph, a chordal graph, or a graph of bounded treewidth.*

To put Theorem 10.5.1 in perspective, recall that by Theorem 10.2.4 no constant-factor approximation for FAIR k-DIVISION UNDER CONFLICTS exists for general graphs, unless $\mathsf{P}=\mathsf{NP}$.

Chapter 11

Final Remarsks to Part II

In this section we give some final thoughts and possible future research directions related to Part II of this thesis.

11.1 Well-Covered Vector Spaces

In Sections 9.2 and 9.4 we developed two general reductions for the problem of computing a well-covering system of a given graph, that is, a system of linear homogeneous equations representing the well-covered vector space of the graph. Using these reductions, we showed that the problem can be solved in polynomial time in the class of fork-free graphs. For the special case of cographs, a faster algorithm was developed.

As a promising avenue for future research, it would be interesting to study the problem in further generalizations of the class of cographs, for example, in the classes considered in [8, 9], including classes of bounded clique-width, in which the well-coveredness property can be recognized in FPT time (with clique-width as the parameter, see [8]). The complexity of computing the well-covered dimension of a graph, as well as the special case of recognizing graphs with positive well-covered dimension also seem to be questions worthy of further consideration.

11.2 Fair Allocation of Indivisible Items

In Chapter 10 we introduced the Fair k-Division Under Conflicts problem and studied it from a computational complexity point of view, with respect to various restrictions on the conflict graph. In particular, we could show that the problem is strongly NP-hard on general bipartite conflict graphs, but can be solved in pseudo-polynomial time on biconvex bipartite graphs, on chordal graphs, on cocomparability graphs, on graphs of bounded treewidth and, more generally, on graphs of bounded clique-width. There are other graph classes sandwiched between the two classes of our results, for which the complexity of Fair k-Division Under Conflicts remains open.

Of particular interest is the following sequence of inclusions: biconvex bipartite $\subseteq$ convex bipartite $\subseteq$ interval bigraph $\subseteq$ chordal bipartite $\subseteq$ bipartite. Outside this chain of inclusions, we pose the complexity of the problem for planar bipartite conflict graphs as another interesting open question.

Recently, Chiarelli et al. [61] developed dynamic programming pseudo-polynomial time algorithms for the classes of convex bipartite graphs and graphs of bounded tree-independence number. The result for the classes of graphs of bounded tree-independence number extends the result presented in this thesis for the classes of graphs of bounded treewidth. Furthermore, the result for the class of convex bipartite graphs extends the result presented in this thesis for convex bipartite conflict graphs, although the algorithm given by Chiarelli et al. [61] relies on a totally different strategy. As mentioned in [61], a natural question that arises in the study of Fair k-Division Under Conflicts is the identification of further graph width parameters leading to pseudo-polynomial algorithms. One such parameter is *thinness* (see [175]) for which the framework of Bonomo and de Estrada [38] can be adapted to Fair k-Division Under Conflicts. In particular, this leads to an alternative pseudo-polynomial-time algorithm for the class of convex bipartite graphs, and even for the more general class of interval bigraphs [207]. This provides a response to a question stated in the above sequence of inclusions, and leaves the class of chordal bipartite graphs as the remaining open case.

Question 11.2.1. What is the complexity of Fair k-Division Under Conflicts for planar bipartite conflict graphs, and for chordal bipartite conflict graphs?

Finally, observe that the Independent Set problem in a sense connects Fair k-Division Under Conflicts and tame graph classes, studied in Chapter 4, since Independent Set can be solved in polynomial time in tame graphs classes and its weighted version corresponds to Fair 1-Division Under Conflicts. An interesting question that arises here is the following.

Question 11.2.2. What is the complexity of Fair k-Division Under Conflicts when the conflict graph belongs to some fixed tame graph class?

We conclude this chapter with a remark about the k-Coloring problem. As observed by Pilipczuk and Rzążewski [200], there exist tame graph classes for which k-Coloring is NP-complete for some fixed k. Indeed, the main result of [200] implies that the class of P_6-free graphs with clique number at most 5 is tame. On the other hand, Huang [143] showed that 5-Coloring is NP-complete for P_6-free graphs, hence, the problem remains NP-complete for P_6-free graphs with clique number at most 5. Although any 5-coloring of a graph represents a solution to Fair 5-Division Under Conflicts, in Fair k-Division Under Conflicts we allow a partial coloring of a graph, so the result by Pilipczuk and Rzążewski cannot be easily adapted to show that for some k Fair k-Division Under Conflicts is strongly NP-hard in some tame graph class.

Bibliography

[1] N. Abbas and L. K. Stewart. Biconvex graphs: ordering and algorithms. *Discrete Applied Mathematics*, 103(1-3):1–19, 2000.

[2] P. Aboulker, P. Charbit, N. Trotignon, and K. Vušković. Vertex elimination orderings for hereditary graph classes. *Discrete Mathematics*, 338(5):825–834, 2015.

[3] T. Abrishami, M. Chudnovsky, C. Dibek, S. Thomassé, N. Trotignon, and K. Vušković. Graphs with polynomially many minimal separators. *J. Comb. Theory, Ser. B*, 152:248–280, 2022.

[4] T. Abrishami, M. Chudnovsky, M. Pilipczuk, P. Rzążewski, and P. Seymour. Induced subgraphs of bounded treewidth and the container method. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1948–1964. [Society for Industrial and Applied Mathematics (SIAM)], Philadelphia, PA, 2021.

[5] V. E. Alekseev. The effect of local constraints on the complexity of determination of the graph independence number. In *Combinatorial-Algebraic Methods in Applied Mathematics*, pages 3–13. Gorky University Press, 1982. In Russian.

[6] V. E. Alekseev. On the number of maximal independent sets in graphs from hereditary classes. *Combinatorial-algebraic methods in discrete optimization, University of Nizhny Novgorod*, pages 5–8, 1991. In Russian.

[7] J. Alman and V. Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. [Society for Industrial and Applied Mathematics (SIAM)], Philadelphia, PA, 2021.

[8] S. R. Alves, K. K. Dabrowski, L. Faria, S. Klein, I. Sau, and U. S. Souza. On the (parameterized) complexity of recognizing well-covered (r, ℓ)-graph. *Theoret. Comput. Sci.*, 746:36–48, 2018.

[9] R. Araújo, E. Costa, S. Klein, R. Sampaio, and U. S. Souza. FPT algorithms to recognize well covered graphs. *Discrete Math. Theor. Comput. Sci.*, 21(1):Paper NO. 3, 15, 2019.

[10] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8:277–284, 1987.

[11] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[12] J. Backer. Separator orders in interval, cocomparability, and AT-free graphs. *Discrete Appl. Math.*, 159(8):717–726, 2011.

[13] E. Balas and C. S. Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.

[14] H.-J. Bandelt and H. M. Mulder. Distance-hereditary graphs. *J. Combin. Theory Ser. B*, 41(2):182–208, 1986.

[15] N. Bansal and M. Sviridenko. The Santa Claus problem. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 31–40. ACM, New York, 2006.

[16] X. Bei, X. Lu, P. Manurangsi, and W. Suksompong. The price of fairness for indivisible goods. *Theory of Computing Systems*, pages 1–25, 2021.

[17] J. Beisegel, M. Chudnovsky, V. Gurvich, M. Milanič, and M. Servatius. Avoidable vertices and edges in graphs: existence, characterization, and applications. *Discrete Appl. Math.*, 309:285–300, 2022.

[18] J. Beisegel, C. Denkert, E. Köhler, M. Krnc, N. Pivač, R. Scheffler, and M. Strehler. On the end-vertex problem of graph searches. *Discrete Math. Theor. Comput. Sci.*, 21(1):20, 2019. Id/No 13.

[19] J. Beisegel, C. Denkert, E. Köhler, M. Krnc, N. Pivač, R. Scheffler, and M. Strehler. The recognition problem of graph search trees. *SIAM J. Discrete Math.*, 35(2):1418–1446, 2021.

[20] R. Belmonte, Y. Otachi, and P. Schweitzer. Induced minor free graphs: Isomorphism and clique-width. *Algorithmica*, 80(1):29–47, 2018.

[21] C. Berge. Stochastic graphs and strongly perfect graphs - a survey. *Southeast Asian Bull. Math.*, 7:16–25, 1983.

[22] C. Berge. Minimax relations for the partial q-colorings of a graph. *Discrete Mathematics*, 74(1-2):3–14, 1989.

[23] E. Berger, K. Choromanski, M. Chudnovsky, J. Fox, M. Loebl, A. Scott, P. Seymour, and S. Thomassé. Tournaments and colouring. *J. Combin. Theory Ser. B*, 103(1):1–20, 2013.

[24] A. Berry and J.-P. Bordat. Separability generalizes Dirac's theorem. *Discrete Appl. Math.*, 84(1-3):43–53, 1998.

[25] A. Berry, J.-P. Bordat, and O. Cogis. Generating all the minimal separators of a graph. *Internat. J. Found. Comput. Sci.*, 11(3):397–403, 2000.

[26] I. Birnbaum, M. Kuneli, R. McDonald, K. Urabe, and O. Vega. The well-covered dimension of products of graphs. *Discuss. Math. Graph Theory*, 34(4):811–827, 2014.

[27] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, volume 56 of *IMA Vol. Math. Appl.*, pages 1–29. Springer, 1993.

[28] J. Błasiok, M. Kamiński, J.-F. Raymond, and T. Trunck. Induced minors and well-quasi-ordering. *J. Combin. Theory Ser. B*, 134:110–142, 2019.

[29] Z. Blázsik, M. Hujter, A. Pluhár, and Z. Tuza. Graphs with no induced C_4 and $2K_2$. *Discrete Math.*, 115(1-3):51–55, 1993.

[30] H. Bodlaender and K. Jansen. On the complexity of scheduling incompatible jobs with unit-times. In *MFCS '93: Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science*, pages 291–300. Springer, 1993.

[31] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[32] H. L. Bodlaender, T. Kloks, and D. Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discrete Math.*, 8(4):606–616, 1995.

[33] H. L. Bodlaender and A. M. C. A. Koster. Safe separators for treewidth. *Discrete Math.*, 306(3):337–350, 2006.

[34] H. L. Bodlaender and A. M. C. A. Koster. Treewidth computations. I. Upper bounds. *Inform. and Comput.*, 208(3):259–275, 2010.

[35] H. L. Bodlaender and U. Rotics. Computing the treewidth and the minimum fill-in with the modular decomposition. *Algorithmica*, 36(4):375–408, 2003.

[36] V. Boncompagni, I. Penev, and K. Vušković. Clique-cutsets beyond chordal graphs. *J. Graph Theory*, 91(2):192–246, 2019.

[37] É. Bonnet, J. Duron, C. Geniet, S. Thomassé, and A. Wesolek. Maximum independent set when excluding an induced minor: $K_1 + tK_2$ and $tC_3 \uplus C_4$. In I. L. Gørtz, M. Farach-Colton, S. J. Puglisi, and G. Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 23:1–23:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[38] F. Bonomo and D. de Estrada. On the thinness and proper thinness of a graph. *Discrete Appl. Math.*, 261:78–92, 2019.

[39] V. Bouchitté and I. Todinca. Minimal triangulations for graphs with "few" minimal separators. In *Algorithms—ESA '98 (Venice)*, volume 1461 of *Lecture Notes in Comput. Sci.*, pages 344–355. Springer, Berlin, 1998.

[40] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in of weakly triangulated graphs. In *STACS 99 (Trier)*, volume 1563 of *Lecture Notes in Comput. Sci.*, pages 197–206. Springer, Berlin, 1999.

[41] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in: grouping the minimal separators. *SIAM J. Comput.*, 31(1):212–232, 2001.

[42] V. Bouchitté and I. Todinca. Listing all potential maximal cliques of a graph. *Theoret. Comput. Sci.*, 276(1-2):17–32, 2002.

[43] S. Bouveret, K. Cechlárová, E. Elkind, A. Igarashi, and D. Peters. Fair division of a graph. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 135–141, 2017.

[44] S. Bouveret, Y. Chevaleyre, and N. Maudet. Fair allocation of indivisible goods. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 284–310. Cambridge University Press, 2016.

[45] A. Brandstädt, J. Engelfriet, H.-O. Le, and V. V. Lozin. Clique-width for 4-vertex forbidden subgraphs. *Theory Comput. Syst.*, 39(4):561–590, 2006.

[46] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: a Survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[47] A. Brandstädt, Van Bang Le, and J. P. Spinrad. *Graph classes: a survey*, volume 3. Philadelphia, PA: SIAM, 1999.

[48] R. C. Brewster, P. Hell, and R. Rizzi. Oriented star packings. *J. Combin. Theory Ser. B*, 98(3):558–576, 2008.

[49] J. Brown and R. Hoshino. Well-covered circulant graphs. *Discrete Math.*, 311(4):244–251, 2011.

[50] J. I. Brown and R. J. Nowakowski. Well-covered vector spaces of graphs. *SIAM J. Discrete Math.*, 19(4):952–965, 2006.

[51] J. I. Brown, R. J. Nowakowski, and I. E. Zverovich. The structure of well-covered graphs with no cycles of length 4. *Discrete Mathematics*, 307(17-18):2235–2245, 2007.

[52] Y. Caro, M. N. Ellingham, and J. E. Ramey. Local structure when all maximal independent sets have equal weight. *SIAM J. Discrete Math.*, 11(4):644–654, 1998.

[53] Y. Caro, A. Sebő, and M. Tarsi. Recognizing greedy structures. *J. Algorithms*, 20(1):137–156, 1996.

[54] Y. Caro and R. Yuster. The uniformity space of hypergraphs and its applications. *Discrete Math.*, 202(1-3):1–19, 1999.

[55] G. J. Chang, A. J. J. Kloks, J. Liu, and S.-L. Peng. The PIGs Full Monty—a floor show of minimal separators. In *STACS 2005*, volume 3404 of *Lecture Notes in Comput. Sci.*, pages 521–532. Springer, Berlin, 2005.

[56] S. Chaplick, M. Töpfer, J. Voborník, and P. Zeman. On H-topological intersection graphs. *Algorithmica*, 83(11):3281–3318, 2021.

[57] L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. P. Gutenberg, and S. Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science—FOCS 2022*, pages 612–623. IEEE Computer Soc., Los Alamitos, CA, 2022.

[58] H. Y. Cheung, T. C. Kwok, and L. C. Lau. Fast matrix rank algorithms and applications. *J. ACM*, 60(5):Art. 31, 25, 2013.

[59] N. Chiarelli, C. Dallard, A. Darmann, S. Lendl, M. Milanič, P. Muršič, U. Pferschy, and N. Pivač. Allocation of indivisible items with individual preference graphs. *Discret. Appl. Math.*, 334:45–62, 2023.

[60] N. Chiarelli, M. Krnc, M. Milanič, U. Pferschy, N. Pivač, and J. Schauer. Fair allocation of indivisible items with conflict graphs. *Algorithmica*, 85(5):1459–1489, 2023.

[61] N. Chiarelli, M. Krnc, M. Milanič, U. Pferschy, and J. Schauer. Fair allocation algorithms for indivisible items under structured conflict constraints. *Comput. Appl. Math.*, 42(7):21, 2023.

[62] N. Chiarelli and M. Milanič. Linear separation of connected dominating sets in graphs. *Ars Math. Contemp.*, 16(2):487–525, 2019.

[63] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Ann. Math. (2)*, 164(1):51–229, 2006.

[64] M. Chudnovsky and P. Seymour. The three-in-a-tree problem. *Combinatorica*, 30(4):387–417, 2010.

[65] M. Chudnovsky, P. D. Seymour, and N. Trotignon. Detecting an induced net subdivision. *J. Comb. Theory, Ser. B*, 103(5):630–641, 2013.

[66] M. Chudnovsky, S. Thomassé, N. Trotignon, and K. Vuskovic. Maximum independent sets in (pyramid, even hole)-free graphs. *CoRR*, abs/1912.11246, 2019.

[67] V. Chvátal and P. L. Hammer. Aggregation of inequalities in integer programming. *Annals of Discrete Mathematics*, 1:145–162, 1977.

[68] V. Chvátal and P. J. Slater. A note on well-covered graphs. In *Quo vadis, graph theory? A source book for challenges and directions*, pages 179–181. Amsterdam: North-Holland, 1993.

[69] F. Cicalese and M. Milanič. Graphs of separability at most 2. *Discrete Appl. Math.*, 160(6):685–696, 2012.

[70] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Universally signable graphs. *Combinatorica*, 17(1):67–77, 1997.

[71] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* MIT Press, Cambridge, MA, third edition, 2009.

[72] D. Corneil, M. Habib, C. Paul, and M. Tedder. A recursive linear time modular decomposition algorithm via lexbfs, 2024.

[73] D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3(3):163–174, 1981.

[74] D. G. Corneil and U. Rotics. On the relationship between clique-width and treewidth. *SIAM Journal on Computing*, 34(4):825–847, 2005.

[75] D. Coudert and G. Ducoffe. Revisiting decomposition by clique separators. *SIAM J. Discrete Math.*, 32(1):682–694, 2018.

[76] D. Coudert, G. Ducoffe, and N. Nisse. To approximate treewidth, use treelength! *SIAM J. Discrete Math.*, 30(3):1424–1436, 2016.

[77] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[78] B. Courcelle and I. Durand. Computations by fly-automata beyond monadic second-order logic. *Theoretical Computer Science*, 619:32–67, 2016.

[79] B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46(2):218–270, 1993.

[80] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[81] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

[82] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms.* Springer, 2015.

[83] K. K. Dabrowski, M. Johnson, G. Paesani, D. Paulusma, and V. Zamaraev. On the price of independence for vertex cover, feedback vertex set and odd cycle transversal. In *43rd International Symposium on Mathematical Foundations of Computer Science*, volume 117 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 63, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.

[84] K. K. Dabrowski, M. Johnson, and D. Paulusma. Clique-width for hereditary graph classes. In *Surveys in combinatorics 2019*, volume 456 of *London Math. Soc. Lecture Note Ser.*, pages 1–56. Cambridge Univ. Press, Cambridge, 2019.

[85] C. Dallard, M. Dumas, C. Hilaire, M. Milanič, A. Perez, and N. Trotignon. Detecting $K_{2,3}$ as an induced minor. In A. Rescigno and U. Vaccaro, editors, *Proceedings of the 35th International Workshop on Combinatorial Algorithms (IWOCA 2024)*, 2024. Lecture Notes in Computer Science, to appear.

[86] C. Dallard, M. Milanič, and K. Štorgel. Treewidth versus clique number. I. Graph classes with a forbidden structure. *SIAM J. Discrete Math.*, 35(4):2618–2646, 2021.

[87] C. Dallard, M. Milanič, and K. Štorgel. Treewidth versus clique number. II. tree-independence number. *J. Comb. Theory, Ser. B*, 164:404–442, 2024.

[88] C. Dallard, M. Milanič, and K. Štorgel. Treewidth versus clique number. III. tree-independence number of graphs with a forbidden structure. *Journal of Combinatorial Theory, Series B*, 167:338–391, 2024.

[89] A. Darmann, U. Pferschy, J. Schauer, and G. J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159:1726–1735, 2011.

[90] D. de Werra. Packing independent sets and transversals. In *Combinatorics and graph theory*, volume 25 of *Banach Center Publ.*, pages 233–240. PWN, Warsaw, 1989.

[91] N. Dean and J. Zito. Well-covered graphs and extendability. *Discrete Math.*, 126(1-3):67–80, 1994.

[92] J. S. Deogun, T. Kloks, D. Kratsch, and H. Müller. On the vertex ranking problem for trapezoid, circular-arc and other graphs. *Discrete Appl. Math.*, 98(1-2):39–63, 1999.

[93] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[94] G. Ding. Subgraphs and well-quasi-ordering. *J. Graph Theory*, 16(5):489–502, 1992.

[95] G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

[96] H. Donkers and B. M. P. Jansen. A Turing kernelization dichotomy for structural parameterizations of $\mathcal{F}$-minor-free deletion. *J. Comput. System Sci.*, 119:164–182, 2021.

[97] M. Dyer, M. Jerrum, H. Müller, and K. Vušković. Counting weighted independent sets beyond the permanent. *SIAM J. Discrete Math.*, 35(2):1503–1524, 2021.

[98] T. Erlebach, H. Kellerer, and U. Pferschy. Multiobjective knapsack problems. *Management Science*, 48:1603–1612, 2002.

[99] W. Espelage, F. Gurski, and E. Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Graph-theoretic concepts in computer science*, volume 2204 of *Lecture Notes in Computer Science*, pages 117–128. Springer, 2001.

[100] L. Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

[101] G. Even, M. M. Halldórsson, L. Kaplan, and D. Ron. Scheduling with conflicts: online and offline algorithms. *Journal of Scheduling*, 12(2):199–224, 2009.

[102] P. Factorovich, I. Méndez-Díaz, and P. Zabala. Pickup and delivery problem with incompatibility constraints. *Computers and Operations Research*, 113:104805, 2020.

[103] O. Favaron. Very well covered graphs. *Discrete Math.*, 42:177–187, 1982.

[104] M. R. Fellows, J. Kratochvíl, M. Middendorf, and F. Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995.

[105] M. R. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider. Clique-width is NP-complete. *SIAM Journal on Discrete Mathematics*, 23(2):909–939, 2009.

[106] A. Finbow, B. Hartnell, and R. J. Nowakowski. A characterization of well covered graphs of girth 5 or greater. *J. Comb. Theory, Ser. B*, 57(1):44–68, 1993.

[107] A. Finbow, B. Hartnell, and R. J. Nowakowski. A characterization of well-covered graphs that contain neither 4- nor 5- cycles. *J. Graph Theory*, 18(7):713–721, 1994.

[108] S. Foldes and P. L. Hammer. Split graphs. *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing (Louisiana State Univ., Baton Rouge, La., 1977)*, pages 311–315. Congressus Numerantium, No. XIX, 1977.

[109] F. V. Fomin and T. Korhonen. Fast FPT-approximation of branchwidth. In *STOC '22—Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 886–899. ACM, New York, 2022.

[110] F. V. Fomin, D. Kratsch, I. Todinca, and Y. Villanger. Exact algorithms for treewidth and minimum fill-in. *SIAM J. Comput.*, 38(3):1058–1079, 2008.

[111] F. V. Fomin, I. Todinca, and Y. Villanger. Large induced subgraphs via triangulations and CMSO. *SIAM J. Comput.*, 44(1):54–87, 2015.

[112] F. V. Fomin and Y. Villanger. Finding induced subgraphs via minimal triangulations. In *STACS 2010: 27th International Symposium on Theoretical Aspects of Computer Science*, volume 5 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 383–394. Wadern: Schloss Dagstuhl – Leibniz Zentrum für Informatik, 2010.

[113] F. V. Fomin and Y. Villanger. Treewidth computation and extremal combinatorics. *Combinatorica*, 32(3):289–308, 2012.

[114] A. Frank. Some polynomial algorithms for certain graphs and hypergraphs. In *Proceedings of the Fifth British Combinatorial Conference (Univ. Aberdeen, Aberdeen, 1975)*, Congressus Numerantium, No. XV, pages 211–226, 1976.

[115] J. Gajarský, L. Jaffke, P. T. Lima, J. Novotná, M. Pilipczuk, P. Rzążewski, and U. S. Souza. Taming graphs with no large creatures and skinny ladders. In *30th Annual European Symposium on Algorithms (ESA 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[116] M. Garey, D. Johnson, G. Miller, and C. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic Discrete Methods*, 1(2):216–227, 1980.

[117] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976.

[118] P. Gartland. *Quasi-Polynomial Time Techniques for Independent Set and Beyond in Hereditary Graph Classes*. PhD thesis, UC Santa Barbara, 2023.

[119] P. Gartland and D. Lokshtanov. Independent set on P_k-free graphs in quasi-polynomial time. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science—FOCS 2020*, pages 613–624. IEEE Computer Soc., Los Alamitos, CA, 2020.

[120] P. Gartland and D. Lokshtanov. Graph classes with few minimal separators. I. Finite forbidden induced subgraphs. In N. Bansal and V. Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3063–3097. SIAM, 2023.

[121] P. Gartland and D. Lokshtanov. Graph classes with few minimal separators. II. A dichotomy. In N. Bansal and V. Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3098–3178. SIAM, 2023.

[122] P. Gartland, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and P. Rzążewski. Finding large induced sparse subgraphs in $C_{>t}$-free graphs in quasipolynomial time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 330–341, New York, NY, USA, 2021. Association for Computing Machinery.

[123] S. Gaspers and S. Mackenzie. On the number of minimal separators in graphs. *J. Graph Theory*, 87(4):653–659, 2018.

[124] M. U. Gerber and D. Kobler. Algorithms for vertex-partitioning problems on graphs with fixed clique-width. *Theoretical Computer Science*, 299(1-3):719–734, 2003.

[125] P. A. Golovach, M. Johnson, D. Paulusma, and J. Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *J. Graph Theory*, 84(4):331–363, 2017.

[126] M. C. Golumbic. Trivially perfect graphs. *Discrete Math.*, 24(1):105–107, 1978.

[127] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, volume 57 of *Annals of Discrete Mathematics*. Elsevier Science B.V., Amsterdam, second edition, 2004.

[128] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics: Study and Research Texts*. Springer-Verlag, Berlin, 1988.

[129] A. Grzesik, T. Klimošová, M. Pilipczuk, and M. Pilipczuk. Covering minimal separators and potential maximal cliques in P_t-free graphs. *Electron. J. Combin.*, 28(1):Paper No. 1.29, 14, 2021.

[130] A. Grzesik, T. Klimošová, M. Pilipczuk, and M. Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *ACM Trans. Algorithms*, 18(1), jan 2022.

[131] F. Gurski. A comparison of two approaches for polynomial time algorithms computing basic graph parameters. *arXiv:0806.4073*, 2008.

[132] M. Habib and C. Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.

[133] T. Hanaka, H. L. Bodlaender, T. C. van der Zanden, and H. Ono. On the maximum weight minimal separator. *Theoret. Comput. Sci.*, 796:294–308, 2019.

[134] T. R. Hartinger. *New Characterizations in Structural Graph Theory: 1-Perfectly Orientable Graphs, Graph Products, and the Price of Connectivity*. PhD thesis, University of Primorska, 2017. `https://www.famnit.upr.si/sl/studij/zakljucna_dela/download/532`.

[135] T. R. Hartinger, M. Johnson, M. Milanič, and D. Paulusma. The price of connectivity for cycle transversals. *European J. Combin.*, 58:203–224, 2016.

[136] T. R. Hartinger and M. Milanič. Partial characterizations of 1-perfectly orientable graphs. *J. Graph Theory*, 85(2):378–394, 2017.

[137] B. L. Hartnell. Well-covered graphs. *J. Combin. Math. Combin. Comput.*, 29:107–115, 1999.

[138] J. Hauschild, J. Ortiz, and O. Vega. On the Levi graph of point-line configurations. *Involve*, 8(5):893–900, 2015.

[139] P. Heggernes. Minimal triangulations of graphs: a survey. *Discrete Mathematics*, 306(3):297–317, 2006.

[140] P. Hell and J. Nešetřil. On the complexity of H-coloring. *J. Combin. Theory Ser. B*, 48(1):92–110, 1990.

[141] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Appl. Math.*, 6:243–254, 1983.

[142] Z.-H. Hu, J.-B. Sheu, L. Zhao, and C.-C. Lu. A dynamic closed-loop vehicle routing problem with uncertainty and incompatible goods. *Transportation Research Part C: Emerging Technologies*, 55:273–297, 2015.

[143] S. Huang. Improved complexity results on k-coloring P_t-free graphs. *Eur. J. Comb.*, 51:336–346, 2016.

[144] M. A. Javidian, M. Valtorta, and P. Jamshidi. AMP chain graphs: minimal separators and structure learning algorithms. *J. Artificial Intelligence Res.*, 69:419–470, 2020.

[145] J. Jeong, E. J. Kim, and S. Oum. Finding branch-decompositions of matroids, hypergraphs, and more. *SIAM Journal on Discrete Mathematics*, 35(4):2544–2617, 2021.

[146] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.

[147] S. Klein, C. P. de Mello, and A. Morgana. Recognizing well covered graphs of families with special P_4-components. *Graphs Combin.*, 29(3):553–567, 2013.

[148] T. Kloks. *Treewidth*, volume 842 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1994.

[149] T. Kloks. Treewidth of circle graphs. *International Journal of Foundations of Computer Science*, 7(02):111–120, 1996.

[150] T. Kloks, H. Bodlaender, H. Müller, and D. Kratsch. Computing treewidth and minimum fill-in: all you need are the minimal separators. In *Algorithms—ESA '93 (Bad Honnef, 1993)*, volume 726 of *Lecture Notes in Comput. Sci.*, pages 260–271. Springer, Berlin, 1993.

[151] T. Kloks, H. Bodlaender, H. Müller, and D. Kratsch. Erratum to esa 1993 proceedings. In J. van Leeuwen, editor, *Algorithms — ESA '94*, pages 508–508, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[152] T. Kloks and D. Kratsch. Finding all minimal separators of a graph. In *STACS 94 (Caen, 1994)*, volume 775 of *Lecture Notes in Comput. Sci.*, pages 759–768. Springer, Berlin, 1994.

[153] T. Kloks, D. Kratsch, and J. Spinrad. On treewidth and minimum fill-in of asteroidal triple-free graphs. *Theoret. Comput. Sci.*, 175(2):309–335, 1997.

[154] T. Kloks, D. Kratsch, and C. K. Wong. Minimum fill-in on circle and circular-arc graphs. *J. Algorithms*, 28(2):272–289, 1998.

[155] E. Köhler. *Graphs without asteroidal triples*. PhD thesis, Technische Universität Berlin, 1999.

[156] T. Korhonen. Finding optimal triangulations parameterized by edge clique cover. *Algorithmica*, 84(8):2242–2270, 2022.

[157] T. Korhonen. Grid induced minor theorem for graphs of small degree. *J. Combin. Theory Ser. B*, 160:206–214, 2023.

[158] T. Korhonen and D. Lokshtanov. Induced-minor-free graphs: Separator theorem, subexponential algorithms, and improved hardness of recognition. In D. P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 5249–5275. SIAM, 2024.

[159] D. Král', J. Kratochvíl, Z. Tuza, and G. J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In *Graph-Theoretic Concepts in Computer Science (Boltenhagen, 2001)*, volume 2204 of *Lecture Notes in Comput. Sci.*, pages 254–262. Springer, Berlin, 2001.

[160] K. Lai, H. Lu, and M. Thorup. Three-in-a-tree in near linear time. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1279–1292. ACM, 2020.

[161] N. Le. Detecting an induced subdivision of K_4. *J. Graph Theory*, 90(2):160–171, 2019.

[162] P. G. H. Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the Association for Computing Machinery*, 21:569–575, 1974.

[163] B. Lévêque, D. Y. Lin, F. Maffray, and N. Trotignon. Detecting induced subgraphs. *Discret. Appl. Math.*, 157(17):3540–3551, 2009.

[164] V. E. Levit and D. Tankus. Weighted well-covered claw-free graphs. *Discrete Mathematics*, 338(3):99–106, 2015.

[165] V. E. Levit and D. Tankus. Well-covered graphs without cycles of lengths 4, 5 and 6. *Discrete Appl. Math.*, 186:158–167, 2015.

[166] M. Loebl and S. Poljak. Efficient subgraph packing. *J. Combin. Theory Ser. B*, 59(1):106–121, 1993.

[167] V. V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595–604, 2008.

[168] V. V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 26–30. ACM, New York, 2006.

[169] F. Maffray and M. Preissmann. Linear recognition of pseudo-split graphs. *Discrete Appl. Math.*, 52(3):307–312, 1994.

[170] F. Maffray, N. Trotignon, and K. Vušković. Algorithms for square-3pc$(\cdot,\cdot)$-free Berge graphs. *SIAM J. Discrete Math.*, 22(1):51–71, 2008.

[171] N. V. R. Mahadev, U. N. Peled, and F. Sun. Equistable graphs. *J. Graph Theory*, 18(3):281–299, 1994.

[172] K. Makino and T. Uno. New algorithms for enumerating all maximal cliques. In *Algorithm theory—SWAT 2004*, volume 3111 of *Lecture Notes in Comput. Sci.*, pages 260–272. Springer, Berlin, 2004.

[173] A. Mallek and M. Boudhar. Scheduling on uniform machines with a conflict graph: complexity and resolution. *Int. Trans. Oper. Res.*, 31(2):863–888, 2024.

[174] D. S. Malyshev. A complexity dichotomy and a new boundary class for the dominating set problem. *J. Comb. Optim.*, 32(1):226–243, 2016.

[175] C. Mannino, G. Oriolo, F. Ricci, and S. Chandran. The stable set problem and the thinness of a graph. *Oper. Res. Lett.*, 35(1):1–9, 2007.

[176] F. Mazoit. *Décomposition algorithmique des graphes*. PhD thesis, École normale supérieure de Lyon, 2004.

[177] R. M. McConnell and J. P. Spinrad. Modular decomposition and transitive orientation. *Discrete Math.*, 201(1-3):189–241, 1999.

[178] T. A. McKee. Requiring that minimal separators induce complete multipartite subgraphs. *Discuss. Math. Graph Theory*, 38(1):263–273, 2018.

[179] T. A. McKee. Graphs that have separator tree representations. *Australas. J. Combin.*, 80:89–98, 2021.

[180] M. Milanič, I. Penev, N. Pivač, and K. Vušković. Bisimplicial separators. *J. Graph Theory*, 2024. doi:10.1002/jgt.23098.

[181] M. Milanič and N. Pivač. Computing well-covered vector spaces of graphs using modular decomposition. *Comput. Appl. Math.*, 42(8):360, 2023.

[182] M. Milanič. *Algorithmic developments and complexity results for finding maximum and exact independent sets in graphs.* PhD thesis, Rutgers University, 2007.

[183] M. Milanič and N. Pivač. Polynomially bounding the number of minimal separators in graphs: reductions, sufficient conditions, and a dichotomy theorem. *Electron. J. Combin.*, 28(1):Paper No. 1.41, 27, 2021.

[184] M. Milanič and N. Pivač. Minimal separators in graph classes defined by small forbidden induced subgraphs. In I. Sau and D. M. Thilikos, editors, *Graph-Theoretic Concepts in Computer Science - 45th International Workshop, WG 2019, Vall de Núria, Spain, June 19-21, 2019, Revised Papers*, volume 11789 of *Lecture Notes in Computer Science*, pages 379–391. Springer, 2019.

[185] P. Montealegre and I. Todinca. On distance-d independent set and other problems in graphs with "few" minimal separators. In *Graph-Theoretic Concepts in Computer Science*, volume 9941 of *Lecture Notes in Comput. Sci.*, pages 183–194. Springer, Berlin, 2016.

[186] A. Muritiba, M. Iori, E. Malaguti, and P. Toth. Algorithms for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 22(3):401–415, 2010.

[187] S. D. Nikolopoulos and L. Palios. Minimal separators in P_4-sparse graphs. *Discrete Math.*, 306(3):381–392, 2006.

[188] S. Olariu. Paw-free graphs. *Inform. Process. Lett.*, 28(1):53–54, 1988.

[189] Y. Otachi and P. Schweitzer. Isomorphism on subgraph-closed graph classes: A complexity dichotomy and intermediate graph classes. In L. Cai, S. Cheng, and T. W. Lam, editors, *Algorithms and Computation - 24th International Symposium, ISAAC 2013, Hong Kong, China, December 16-18, 2013, Proceedings*, volume 8283 of *Lecture Notes in Computer Science*, pages 111–118. Springer, 2013.

[190] S. Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms*, 5(1):Art. 10, 20, 2009.

[191] S. Oum and P. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory. Series B*, 96(4):514–528, 2006.

[192] D. Pálvölgi. Partitioning to three matchings of given size is NP-complete for bipartite graphs. *Acta Universitatis Sapientiae, Informatica*, 6(2):206–209, 2014.

[193] A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Appl. Math.*, 79(1-3):171–188, 1997.

[194] C. Payan. A class of threshold and domishold graphs: equistable and equidominating graphs. *Discrete Math.*, 29(1):47–52, 1980.

[195] V. Pedrotti and C. P. de Mello. Minimal separators in P_4-tidy graphs. In *LAGOS'09—V Latin-American Algorithms, Graphs and Optimization Symposium*, volume 35 of *Electron. Notes Discrete Math.*, pages 71–76. Elsevier Sci. B. V., Amsterdam, 2009.

[196] V. Pedrotti and C. P. de Mello. Minimal separators in extended P_4-laden graphs. *Discrete Appl. Math.*, 160(18):2769–2777, 2012.

[197] U. Pferschy and J. Schauer. The knapsack problem with conflict graphs. *Journal of Graph Algorithms and Applications*, 13(2):233–249, 2009.

[198] U. Pferschy and J. Schauer. Approximation of knapsack problems with conflict and forcing graphs. *Journal of Combinatorial Optimization*, 33(4):1300–1323, 2017.

[199] M. Pilipczuk, M. Pilipczuk, and P. Rzążewski. Quasi-polynomial-time algorithm for independent set in P_t-free graphs via shrinking the space of induced paths. In H. V. Le and V. King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 204–209. [Society for Industrial and Applied Mathematics (SIAM)], Philadelphia, PA, 2021.

[200] M. Pilipczuk and P. Rzążewski. A polynomial bound on the number of minimal separators and potential maximal cliques in P_6-free graphs of bounded clique number. *arXiv:2310.11573*, 2023.

[201] M. D. Plummer. Some covering concepts in graphs. *J. Comb. Theory*, 8:91–98, 1970.

[202] M. D. Plummer. Well-covered graphs: a survey. *Quaestiones Math.*, 16(3):253–287, 1993.

[203] S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974.

[204] M. Pouzet, H. Si Kaddour, and N. Trotignon. Claw-freeness, 3-homogeneous subsets of a graph and a reconstruction problem. *Contrib. Discrete Math.*, 6(1):86–97, 2011.

[205] E. Prisner. Graphs with few cliques. In *Graph theory, combinatorics, and algorithms, Vol. 1, 2 (Kalamazoo, MI, 1992)*, Wiley-Intersci. Publ., pages 945–956. Wiley, New York, 1992.

[206] E. Prisner, J. Topp, and P. D. Vestergaard. Well covered simplicial, chordal, and circular arc graphs. *J. Graph Theory*, 21(2):113–119, 1996.

[207] A. Rafiey. Recognizing interval bigraphs by forbidden patterns. *J. Graph Theory*, 100(3):504–529, 2022.

[208] V. Raghavan and J. P. Spinrad. Robust algorithms for restricted domains. *J. Algorithms*, 48(1):160–172, 2003.

[209] F. P. Ramsey. On a Problem of Formal Logic. *Proc. London Math. Soc. (2)*, 30(4):264–286, 1929.

[210] M. Rao. MSOL partitioning problems on graphs of bounded treewidth and clique-width. *Theoretical Computer Science*, 377(1-3):260–267, 2007.

[211] G. Ravindra. Well-covered graphs. *J. Comb. Inf. Syst. Sci.*, 2:20–21, 1977.

[212] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976.

[213] N. D. Roussopoulos. A max $\{m, n\}$ algorithm for determining the graph H from its line graph G. *Information Processing Letters*, 2:108–112, 1973.

[214] R. S. Sankaranarayana and L. K. Stewart. Complexity results for well-covered graphs. *Networks*, 22(3):247–262, 1992.

[215] L. F. M. Santos, R. S. Iwayama, L. B. Cavalcanti, L. M. Turi, F. E. de Souza Morais, G. Mormilho, and C. B. Cunha. A variable neighborhood search algorithm for the bin packing problem with compatible categories. *Expert Systems with Applications*, 124:209–225, 2019.

[216] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency.*, volume 24 of *Algorithms and Combinatorics.* Springer, 2003.

[217] P. Schweitzer. Towards an isomorphism dichotomy for hereditary graph classes. *Theory Comput. Syst.*, 61(4):1084–1127, 2017.

[218] K. Skodinis. Efficient analysis of graphs with small minimal separators. In *Graph-Theoretic Concepts in Computer Science*, volume 1665 of *Lecture Notes in Comput. Sci.*, pages 155–166. Springer, Berlin, 1999.

[219] J. Spinrad. On comparability and permutation graphs. *SIAM Journal on Computing*, 14(3):658–670, 1985.

[220] J. P. Spinrad. *Efficient Graph Representations*, volume 19 of *Fields Institute Monographs.* American Mathematical Society, Providence, RI, 2003.

[221] K. Suchan. *Minimal Separators in Intersection Graphs.* Master's thesis, Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, 2003.

[222] J. J. Sylvester. Chemistry and algebra. *Nature*, 17(432):284, Feb. 1878.

[223] J. J. Sylvester. On Subvariants, i.e. Semi-Invariants to Binary Quantics of an Unlimited Order. *Amer. J. Math.*, 5(1-4):79–136, 1882.

[224] D. Tankus and M. Tarsi. Well-covered claw-free graphs. *J. Comb. Theory, Ser. B*, 66(2):293–302, 1996.

[225] D. Tankus and M. Tarsi. The structure of well-covered graphs and the complexity of their recognition problems. *J. Comb. Theory, Ser. B*, 69(2):230–233, 1997.

[226] R. E. Tarjan. Decomposition by clique separators. *Discrete Math.*, 55(2):221–232, 1985.

[227] W. Thomson. Introduction to the theory of fair allocation. In F. Brandt et al., editor, *Handbook of Computational Social Choice*, chapter 11, pages 261–283. Cambridge University Press, 2016.

[228] N. Trotignon and L. A. Pham. χ-bounds, operations, and chords. *Journal of Graph Theory*, 88(2):312–336, 2018.

[229] A. Tucker. A structure theorem for the consecutive 1's property. *Journal of Combinatorial Theory. Series B*, 12:153–162, 1972.

[230] S. Wagon. A bound on the chromatic number of graphs without certain induced subgraphs. *J. Combin. Theory Ser. B*, 29(3):345–346, 1980.

[231] V. V. Williams, J. R. Wang, R. Williams, and H. Yu. Finding four-node subgraphs in triangle time. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1671–1680. SIAM, Philadelphia, PA, 2015.

[232] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Algebraic Discrete Methods*, 2:77–79, 1981.

[233] Y. Ye and A. Borodin. Elimination graphs. *ACM Transactions on Algorithms*, 8(2):Article 14, 1–23, 2012.

[234] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007.

Povzetek v slovenskem jeziku

Osrednja tema doktorske disertacije sodi na področje diskretne matematike, oz. natančneje, na področje teorije grafov. Graf G je matematična struktura, ki se uporablja za modeliranje binarnih relacij med objekti in sestoji iz množice točk $V = V(G)$ in množice povezav $E = E(G)$, pri čemer so povezave neurejeni pari točk. Veliko praktičnih problemov lahko predstavimo z grafi. Tako grafe lahko uporabimo za modeliranje različnih odnosov in procesov v fizikalnih, bioloških, družbenih in informacijskih sistemih. Prvi članek v zgodovini teorije grafov je leta 1736 objavil Leonhard Euler [100], medtem ko je izraz "graf" uvedel Sylvester v članku, objavljenem leta 1878 [222].

Dve točki a in b v grafu G sta *sosednji* v G, če obstaja povezava $ab \in E(G)$, ki ju povezuje; v tem primeru rečemo, da sta a in b *soseda* v G. Množici vseh sosedov točke v v grafu G rečemo *soseščina* točke v v G. Z leti so bili predstavljeni in študirani različni koncepti v grafih. Za disertacijo sta še zlasti pomembna koncepta minimalnega separatorja in neodvisne množice. *Minimalen separator* v grafu G je taka minimalna množica točk grafa, katere odstranitev povzroči, da dve fiksni točki grafa postaneta nepovezani. *Neodvisna množica* v grafu G je množica paroma nesosednjih točk. Problem iskanja največje take množice v grafu je znan NP-težek problem [146].

V doktorski disertaciji študiramo minimalne separatorje in neodvisne množice v grafih. S tem namenom je disertacija razdeljena na dva dela. Prvi del je namenjen študiju minimalnih separatorjev, drugi pa obravnava nekatere stare in nove algoritmične probleme, povezane s problemom neodvisnih množic. Osrednja skupna tema obeh delov je študija razredov grafov, tj. množic grafov, zaprtih za izomorfizem.

Prvi del

V prvem delu doktorske disertacije se osredotočimo na študij minimalnih separatorjev v grafih. Za graf G in dve nesosednji točki a in b v G rečemo, da je množica $S \subseteq V(G) \setminus \{a, b\}$ (a, b)*-separator*, če sta točki a and b vsebovani v različnih povezanih komponentah grafa $G - S$. Če množica S ne vsebuje nobenega drugega (a, b)-separatorja kot pravo podmnožico, potem je S *minimalen* (a, b)-separator. *Minimalen separator* v grafu G je vsaka množica $S \subseteq V(G)$, ki je minimalen (a, b)-separator za nek par nesosednjih točk a in b. Pri tem je možno, da je S minimalen separator grafa G, čeprav je neka množica $S' \subsetneqq S$ tudi separator grafa G. Dejansko lahko obstaja tak par a, b nesosednjih točk, da je S minimalen (a, b)-separator v G, pa tudi kakšen tak drug par a', b' nesosednjih točk, da je nek $S' \subsetneqq S$ (a', b')-separator v G.

Graf je *tetiven*, če ne vsebuje induciranih ciklov dolžine vsaj štiri. Študije minimalnih separatorjev v grafih so se začele vsaj v šestdesetih let 20. stoletja, ko so bili tetivni grafi karakterizirani kot natanko tisti grafi, v katerih so vsi minimalni separatorji klike [95]. Minimalni separatorji so bili kasneje študirani v [24] v kontekstu mopleksov, igrali so pomembno vlogo pri računanju z redkimi matrikami z uporabo minimalnih triangulacij (glej npr. pregledni članek [139]) in imeli so tudi številne uporabe pri razvoju algoritmov (glej npr. [12, 33, 41, 226]). Posledično danes obstajajo številni algoritmi in karakterizacije grafov, ki temeljijo na minimalnih separatorjih (glej npr. [12, 24, 33, 34, 35, 41, 62, 76, 95, 153, 193]). V doktorski disertaciji obravnavamo tri medsebojno povezane probleme o minimalnih separatorjih. Preden jih predstavimo, potrebujemo nekaj definicij.

Graf F je *induciran podgraf* grafa G, če velja $V(F) \subseteq V(G)$ in $E(F) = \{uv \in E(G) \mid \{u,v\} \subseteq V(F)\}$. V tem primeru rečemo, da je *F podgraf grafa G, induciran z $V(F)$*. Za množico $S \subseteq V(G)$ označimo z $G[S]$ podgraf grafa G, induciran z množico S. Če sta F in G grafa, za katera velja, da noben induciran podgraf grafa G ni izomorfen F, potem za graf G pravimo, da je *F-prost*. Če je $\mathcal{F}$ družina grafov, rečemo, da je graf G *$\mathcal{F}$-prost*, če noben induciran podgraf grafa G ni izomorfen nobenemu grafu iz družine $\mathcal{F}$. *Krčenje povezave* $e = uv$ v grafu G je operacija zamenjave točk u in v v G z novo točko w, ki je sosednja natanko vsem točkam v $(N_G(u) \cup N_G(v)) \setminus \{u,v\}$; dobljeni graf je označen z G/e. *Minor* grafa G je graf, dobljen iz G z zaporedjem brisanja točk, brisanja povezav in krčenja povezav. *Induciran minor* grafa G je poljuben graf, dobljen iz G z operacijama brisanja točk in krčenja povezav. *Subdivizija* grafa G je vsak graf, dobljen z zaporednim ponavljanjem operacije 'vstavi točko v povezavo': zamenjaj povezavo uv s povezavami uw in wv, kjer je w nova točka. *Induciran topološki minor* grafa G je poljuben graf H, za katerega velja, da je neka subdivizija grafa H induciran podgraf grafa G. Če graf H ni izomorfen nobenemu induciranemu minorju (ali induciranemu topološkemu minorju) grafa G, potem rečemo, da graf G *ne vsebuje grafa H kot induciran minor* (oziroma *kot induciran topološki minor*). Za več informacij o osnovnih konceptih v grafih bralca napotujemo na [93].

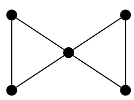
Številni algoritmi na grafih (glej, npr., [42, 112, 185]) v nekem koraku svojega izvajanja enumerirajo vse minimalne separatorje vhodnega grafa. Število minimalnih separatorjev grafa torej neposredno vpliva na čas izvajanja takih algoritmov. Fomin idr. so pokazali v [110], da je največje število minimalnih separatorjev v poljubnem n-točkovnem grafu navzdol omejeno z $\Omega(3^{n/3})$ in navzgor omejeno z $\mathcal{O}(1{,}708^n)$. Kasneje so to mejo neodvisno izboljšali Fomin in Villanger [113] ter Gaspers in Mackenezie [123]. Pokazali so, da je največje število minimalnih separatorjev v grafu navzgor omejeno z $\mathcal{O}(\rho^n n)$, kjer je $\rho = (1+\sqrt{5})/2$. Izkaže se, da za razrede grafov, v katerih je število minimalnih separatorjev omejeno z nekim polinomom, algoritmi iz [42, 112, 185] delujejo v polinomskem času. Veliko problemov, ki so NP-težki za splošne grafe tako postane rešljivih v polinomskem času za razrede grafov s polinomsko omejenim številom minimalnih separatorjev. To velja za Drevesno širino in Minimalno dopolnitev [42], za Največjo neodvisno množico, Povratno množico točk in, splošneje, za problem iskanja največjega induciranega podgrafa, katerega drevesna širina je največ neka konstanta t [112], ter za d-Razdaljno neodvisno množico, za sodo število d [185].

Rezultat Fomina in Villangerja iz [112] so leta 2015 posplošili Fomin, Todinca in Villanger [111], tako da so razvili algoritmičen metaizrek o induciranih podgrafih z lastnostmi, ki jih je mogoče izraziti v določenem logičnem sistemu. Njihov pristop zajema številne probleme, vključno z Največjim induciranim prirejanjem, Najdaljšo inducirano potjo, Največjim induciranim podgrafom brez ciklov dolžine 0 po modulu m, kjer je m poljubno fiksno pozitivno celo število, in Največjim induciranim podgrafom brez minorja iz $\mathcal{F}$, kjer je $\mathcal{F}$ katerakoli množica grafov, ki vsebuje nek ravninski graf.

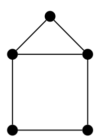
Zaradi vseh teh rezultatov je pomembno identificirati razrede grafov s polinomsko omejenim številom minimalnih separatorjev. Znani razredi s to lastnostjo vključujejo tetivne grafe [212] in njihove posplošitve, šibko tetivne grafe [41], permutacijske grafe [32, 148] in bolj splošno neprimerljivostne grafe omejene intervalne dimenzije [92], grafe krožnih lokov [154], krožne grafe [149], poligonske krožne grafe [221], razdaljno-hereditarne grafe [55, 150], grafe, za katere niti graf, niti njegov komplement ne vsebujeta asteroidalne trojice [155], P_4-redke grafe [187, 196] in grafe z minimalnimi separatorji omejene velikosti [218]. Poleg tega je znano, da ima poljuben razred grafov polinomsko omejeno število minimalnih separatorjev natanko takrat, ko ima polinomsko omejeno število potencialnih maksimalnih klik [42].

Krotki razredi grafov. Zgoraj našteti rezultati nas pripeljejo do prvega ključnega vidika razredov grafov, študiranega v tej doktorski disertaciji. Razredu grafov pravimo, da je *krotek*, če imajo grafi v razredu polinomsko omejeno število minimalnih separatorjev. Natančneje, razred grafov $\mathcal{G}$ je *krotek*, če obstaja tak polinom $p : \mathbb{R} \to \mathbb{R}$, da za vsak graf $G \in \mathcal{G}$ velja $s(G) \le p(|V(G)|)$, kjer je $s(G)$ število minimalnih separatorjev v G. Razredu grafov $\mathcal{G}$ pravimo, da je *divji*, če obstaja tako število $c > 1$, da za vsako poljubno veliko število n obstaja n-točkoven graf v razredu, ki ima c^n minimalnih separatorjev.

K znanju o krotkih razredih grafov smo prispevali iz nekaj medsebojno povezanih zornih kotov. Najprej smo analizirali operacije na grafih, ki ohranjajo krotke razrede grafov. Pokazali smo, da je hereditaren razred grafov $\mathcal{G}$ krotek natanko takrat, ko je podrazred, sestavljen iz grafov v $\mathcal{G}$ brez prereznih klik, krotek (izrek 4.2.3). Podali smo primere grafov z eksponentno mnogo minimalnimi separatorji in pregledali literaturo ter povzeli znane družine grafov, ki niso krotki. S tem smo identificirali potrebne pogoje, ki naj bi jih izpolnjeval vsak krotek razred grafov. Vsak tak pogoj nam da neko neskončno družino krotkih razredov grafov. Ta rezultat in Ramseyjev izrek vodita do več vrst zadostnih pogojev za to, da je razred grafov krotek. Omenjeni rezultati vodijo do dihotomij, ki loči krotke razrede grafov od nekrotkih znotraj družin razredov grafov, definiranih s seznamom prepovedanih grafov glede na določeno relacijo vsebovanosti. V razdelku 4.4 smo karakterizirali krotke razrede grafov v družini razredov grafov, definirane s prepovedanimi induciranimi podgrafi na največ štirih točkah (izrek 4.4.21). V razdelku 4.5 smo karakterizirali krotke razrede grafov v družini razredov grafov, definiranih s posameznim prepovedanim induciranim minorjem, oziroma induciranim topološkim minorjem (izreka 4.5.6 in 4.5.7). Študirali smo prepoznavanje določenih razredov grafov dobljenih v teh raziskavah, in pokazali, da obstajajo algoritmi, ki v polinomskem času prepoznajo grafe, ki ne vsebuje hiše kot induciran topološki minor ali metulja kot induciran minor (glej spodnjo sliko).



metulj



hiša

Rezultati predstavljeni v izrekih 4.4.21, 4.5.6 in 4.5.7 so na ta način prispevali k seznamu dihotomij v teoriji grafov. V literaturi obstajajo podobni rezultati za številne probleme z različnih področjih matematike, vključno s teorijo grafov. To vključuje dihotomije, povezane z omejeno klično širino [84], omejenim kromatičnim številom usmerjenih grafov [23], ceno povezanosti in neodvisnosti [83, 135] ter računsko zahtevnostjo številnih algoritmčnih problemov, kot so HOMOMORFIZEM GRAFA [140], IZOMORFIZEM GRAFA [217], DOMINANTNA MNOŽICA [174] in različni problemi barvanja grafov [125] ter problemi pakiranja [48, 166].

Kot odprta vprašanja smo postavili karakterizacijo krotkih grafov z enim samim prepovedanim minorjem. Drugo odprto vprašanje pa predstavlja posplošitev naših rezultatov, v smislu karakterizacije krotkih grafov s poljubno družino prepovedanih induciranih minorjev ali induciranih topoloških minorjev.

Ekstremalno število minimalnih separatorjev. Med grafi, ki imajo polinomsko omejeno število minimalnih separatorjev, so posebej zanimivi grafi, v katerih je to število linearno in so posledično omenjeni algoritmi na teh grafih posebej učinkoviti. Na primer, n-točkovni razcepljeni grafi nimajo več kot n minimalnih separatorjev [196] in, splošneje, enako velja tudi za tetivne grafe [212] in $2P_2$-proste grafe. Nikolopoulos in Palios sta za n-točkovne kografe in, bolj splošno, n-točkovne P_4-redke grafe določila zgornjo mejo števila minimalnih separatorjev, ki znaša $2n/3$ [187] .

V doktorski disertaciji smo obravnavali ekstremalno vprašanje izračuna največjega števila minimalnih separatorjev v n-točkovnem grafu iz danega razreda, za več medseboj povezanih razredov grafov z največ linearnim številom minimalnih separatorjev: pragovni grafi, razcepljeni grafi, kografi, trivialno popolni grafi in njihovi komplementi, psevdo-razcepljeni grafi in $2P_2$-prosti grafi. Omenjeni razredi grafov so bili študirani iz različnih zornih kotov (glej [46]) in dopuščajo različne karakterizacije; zlasti jih je mogoče vse definirati z majhno množico prepovedanih induciranih podgrafov, ki je podmnožica množice $\{2P_2, P_4, C_4, C_5\}$, pri čemer je P_n pot na n točkah, in C_n cikel na n točkah.

Če imamo podan razred grafov $\mathcal{G}$ in pozitivno celo število n, označimo z $f_{\mathcal{G}}(n)$ največje število minimalnih separatorjev po vseh n-točkovnih grafih $G \in \mathcal{G}$ (pri čemer $f_{\mathcal{G}}(n) = 0$, če $\mathcal{G}$ ne vsebuje n-točkovnega grafa). Če je $\mathcal{G}$ razred vseh grafov, potem je $f_{\mathcal{G}}(n)$ enako $\mathcal{O}\left(((1+\sqrt{5})/2)^n \cdot n\right)$ [113, 123]. V disertaciji zračunamo točno vrednost $f_{\mathcal{G}}(n)$ za vseh sedem zgoraj naštetih razredov grafov in za vse vrednosti števila n. Dobljeni rezultati so predstavljeni v tabeli spodaj in podrobno v razdelku 5. Funkciji a in b v desnem stolpcu tabele zadoščata naslednjim pogojem: $a : \mathbb{N} \to \{0, 1\}$ in $b : \mathbb{N} \to \{-1, 0, 1, 2\}$. Označimo $\log n = \log_2 n$.

Dobljeni rezultati so predstavljeni v naslednji tabeli.

Razredi grafov $\mathcal{G}$	Prepovedani ind. podgrafi	$f_{\mathcal{G}}(n)$
pragovni grafi [67]	$\{2P_2, P_4, C_4\}$	$\lceil (n-1)/2 \rceil$
trivialno popolni grafi [126]	$\{P_4, C_4\}$	$\lceil (n-1)/2 \rceil$
ko-trivialno popolni grafi	$\{2P_2, P_4\}$	$\lceil 2n/3 \rceil - 1$
kografi [73]	$\{P_4\}$	$\lceil 2n/3 \rceil - 1$
razcepljeni grafi [108]	$\{2P_2, C_4, C_5\}$	$n - \lfloor \log n \rfloor - a(n)$
psevdo-razcepljeni grafi [169]	$\{2P_2, C_4\}$	$n - \lfloor \log n \rfloor + b(n)$
$2P_2$-prosti grafi	$\{2P_2\}$	n

Študijo predstavljeno v 5. poglavju lahko nadaljujemo tako, da določimo $f_{\mathcal{G}}(n)$ za določeno izbiro razreda $\mathcal{G}$. Ta koncept pa lahko tudi posplošimo, z uvedbo razreda grafov, v katerem je število minimalnih separatorjev vsakega grafa iz razreda omejeno navzgor s številom točk grafa. Znotraj tega razreda so posebej zanimivi grafi, v katerih obstaja točka, ki jo lahko zbrišemo, pri čemer se število minimalnih separatorjev v grafu zmanjša največ za ena. Posledično lahko definiramo tako eliminacijsko shemo točk, da se število minimalnih separatorjev na vsakem koraku brisanja točk zmanjša za največ ena. Pokažemo lahko da npr. univerzalne, izolirane in simplicialne točke zadoščajo tej lastnosti (glej poglavje 7). Želeli bi pa karakterizirati vse take točke, ali študirati minimalne primere grafov, v katerih take točke ne obstajajo.

Bisimplicialni separatorji. Spomnimo, da so tetivni grafi karakterizirani kot natanko tisti grafi, v katerih so vsi minimalni separatorji klike [95]. V poglavju 6 smo posplošili ta koncept tako, da smo študirali grafe, v katerih lahko vsak minimalen separator izrazimo kot unijo največ k klik, za neko nenegativno celo število k. V splošnem, za dan razred grafov $\mathcal{C}$, označimo z $\mathcal{G}_{\mathcal{C}}$ razred vseh grafov G, za katere velja, da vsak minimalen separator v G inducira graf iz $\mathcal{C}$. Polni grafi nimajo separatorjev, ter za vsako izbiro razreda $\mathcal{C}$, velja, da razred $\mathcal{G}_{\mathcal{C}}$ vsebuje vse polne grafe. Za nenegativno celo število k označimo z $\mathcal{G}_k$ razred vseh grafov G, ki imajo lastnost, da je vsak minimalen separator v G k-simplicialen, tj. unija k (lahko praznih) klik. Če je $k = 1$ (ali $k = 2$), terminologijo poenostavimo, in termin *k-simplicialen* nadomestimo s *simplicialen* (oziroma *bisimplicialen*). Očitno je $\mathcal{G}_0 \subseteq \mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \ldots$. Opazimo, da je $\mathcal{G}_0$ razred vseh disjunktnih unij polnih grafov in (kot sledi iz [95]) $\mathcal{G}_1$ razred vseh tetivnih grafov.

V disertaciji smo študirali grafovske razrede oblike $\mathcal{G}_{\mathcal{C}}$, kjer je $\mathcal{C}$ hereditaren razred grafov, tj. razred grafov, zaprt za operacijo brisanja točk. Poseben poudarek smo dali razredom $\mathcal{G}_k$, še posebej razredu $\mathcal{G}_2$. Kot omenjeno, je razred $\mathcal{G}_2$ razred vseh grafov, ki imajo bisimplicialne separatorje, in vsebuje vse tetivne grafe. Poleg tega je enostavno videti, da je razred vseh grafov krožnih lokov (to so presečni grafi lokov na krogu) vsebovan v razredu $\mathcal{G}_2$. To motivira tako študijo zahtevnosti posameznih algoritmičnih problemov na grafih v razredu $\mathcal{G}_k$, za $k \geq 2$, kot tudi študijo strukture grafov, ki pripadajo $\mathcal{G}_k$, s posebnim poudarkom na primeru $k = 2$.

Študirali smo vpliv različnih operacij grafov na strukturo grafov v razredu $\mathcal{G}_k$, za poljubno nenegativno celo število k. Posebej zanimiva je operacija krčenja povezav, ki v kombinaciji z brisanjem točk vodi do relacije vsebovanosti induciranega minorja. Za vsak $k \geq 0$ smo karakterizirali družino $\mathcal{F}$, za katero velja, da je vsak graf iz $\mathcal{F}$ prepovedan za $\mathcal{G}_k$ kot induciran minor (izrek 6.2.1). Kot posledico smo opisali seznam grafov, ki so minimalni prepovedani inducirani minorji za razred $\mathcal{G}_2$ (posledica 6.2.8). Zatem smo pokazali, da je vsaka LexBFS razvrstitev točk grafa v $\mathcal{G}_k$ v bistvu k-simplicialna razvrstitev točk istega grafa (izrek 6.3.2). Ta rezultat predstavlja posplošitev rezultata za tetivne grafe od Rosa, Tarjana, in Luekerja [212], kar ustreza primeru $k = 1$ našega rezultata. Posledično, pokazali smo, da za vsako pozitivno celo število k vsak neničeln graf v $\mathcal{G}_k$ vsebuje k-simplicialno točko (to je, točko v G, katere soseščina v grafu G je unija k klik), glej Posledico 6.3.4.

V razdelku 6.4 smo pokazali, da je prepoznavanje grafov v $\mathcal{G}_k$ za $k \geq 3$ NP-težko. Prepoznavanje grafov v $\mathcal{G}_2$ je odprt problem. Posledično smo študirali podrazrede razreda $\mathcal{G}_2$, in sicer: preseke razreda $\mathcal{G}_2$ z grafi omejenega kličnega števila, popolnimi grafi in grafi brez diamantov (glej poglavje 6.5). Za le-te smo dobili strukturne karakterizacije in posledično polinomske algoritme za prepoznavanje. Kot posledico zgornjih strukturnih rezultatov smo dobili polinomske algoritme, ki rešijo določene grafovske probleme v omenjenih razredih grafov. Računske zahtevnosti določenih problemov so povzete v tabeli spodaj (število točk in povezav vhodnega grafa sta označena z n in m, z $\omega < 2.3728596$ pa označimo eksponent množenja matrik (glej [7]).

	grafi brez diamantov grafi v $\mathcal{G}_2$	$\mathcal{G}_2$	$\mathcal{G}_k$ $(k \geq 3)$
prepoznavanje	$\mathcal{O}(n^\omega \log n)$	?	NP-težko
Najtežja klika	$\mathcal{O}(n^\omega \log n)$	$\mathcal{O}(n^{3+o(1)})$	NP-težko
Najtežja neodvisna množica	$\mathcal{O}(n^2(n+m))$	$\mathcal{O}(n^5)$	$\mathcal{O}(n^{2k+2})$
Točkovno barvanje	$\mathcal{O}(n^\omega \log n)$	NP-težko	NP-težko

Drugi del

Ko imamo podan graf G in celo število k, lahko definiramo problem Neodvisne Množice kot odločitven problem, ali v grafu G obstaja neodvisna množica moči k. Neodvisna množica je znan NP-poln problem [146]. Neodvisna množica v grafu G je *maksimalna*, če ji ne moremo dodati nobene točke, ne da bi s tem izgubili pogoj neodvisnosti, in *največja*, če v grafu G ni neodvisne množice, ki ima večjo moč. V tem delu doktorske disertacije smo obravnavali dva različna problema, povezana s problemom neodvisne množice. Prvič, študirali smo problem izračuna vektorskega prostora, ki sestoji iz vseh utežnih funkcij na točkah grafa, glede na katere imajo vse maksimalne neodvisne množice grafa enako težo. Drugi problem je problem poštene razdelitve: problem izračuna optimalne razdelitve nedeljivih predmetov agentom, pri čemer upoštevamo konflikten graf in določene kriterije poštene razdelitve. S konfliktnim grafom lahko prepovemo hkratno uporabo določenih predmetov, tako da vsaka množica predmetov, ki je dodeljena določenemu agentu, predstavlja neodvisno množico konfliktnega grafa.

Vektorski prostori dobrega pokritja. Graf je *dobro pokrit*, če imajo vse njegove maksimalne neodvisne množice isto moč. Dobro pokrite grafe je uvedel Plummer leta 1970 [201] in v literaturi so bili obsežno študirani (glej [137] za uvod in [202] za pregled področja). Če vsaki točki grafa G pripišemo realno število, to je *utež* točke, govorimo o *uteženem grafu*. Teža množice $S \subseteq V(G)$ v uteženem grafu G, z utežno funkcijo $w : V(G) \to \mathbb{R}$, je definirana kot $w(S) = \sum_{v \in S} w(v)$. Utežen graf G z utežno funkcijo $w : V(G) \to \mathbb{R}$ je *w-dobro-pokrit*, če so vse maksimalne neodvisne množice v grafu G enake teže glede na utežno funkcijo w. Koncept w-dobro-pokritih grafov so predstavili Caro, Ellingham in Ramey leta 1998 [52] v bolj splošnem kontekstu utežnih funkcij, ki preslikajo točke grafa v elemente abelske grupe (glej tudi [50]).

Dobro pokritje grafa G je poljubna taka realna utežna funkcija w na $V(G)$, za katero velja, da je G w-dobro-pokrit. Znano je, da množica vseh dobrih pokritij grafa G, ki jo označimo z $\mathrm{WCW}(G)$, tvori vektorski prostor nad poljem realnih števil (glej [50, 54]). Imenujemo ga *dobro pokrit prostor* grafa G. Vsak sistem linearnih enačb, ki opiše vektorski prostor $\mathrm{WCW}(G)$, bo poimenovan *sistem dobrega pokritja* grafa G. V doktorski disertaciji smo študirali naslednji problem.

SISTEM DOBREGA POKRITJA **Input:** Graf $G = (V, E)$. **Task:** Izračunaj sistem dobrega pokritja grafa G.

Graf G je dobro pokrit natanko takrat, ko utežna funkcija, ki je konstantno enaka 1, pripada dobro pokritemu prostoru grafa G. Problem prepoznavanja dobro pokritih grafov je co-NP-poln problem (glej [68, 214]) in posledično je tudi bolj splošen problem SISTEM DOBREGA POKRITJA co-NP-težek. Dobro pokrite prostore grafov, ki ne vsebujejo ciklov dolžine 4, so študirali Brown, Nowakowski in Zverovich [51]. SISTEM DOBREGA POKRITJA je mogoče rešiti v polinomskem času v razredih grafov z omejeno maksimalno stopnjo točke, kot so pokazali Caro, Ellingham in Ramey [52], v razredu grafov s premerom najmanj 7, kot sta pokazala Caro in Yuster [54], v razredu grafov brez krempljev, kot sta pokazala Levit in Tankus [164] (kjer je *krempelj* graf na štirih točkah, pri čemer ima ena točka stopnjo 3, tri točke pa stopnjo 1) ter v razredu grafov brez ciklov dolžin 4, 5 in 6 [165]. V disertaciji smo posplošili rezultat dobljen v [164] tako, da smo pokazali, da omenjeni problem lahko rešimo v polinomskem času v razredu grafov brez vilic (pri čemer je *vilica* graf z množico točk $\{v_1, v_2, v_3, v_3, v_5\}$ in množico povezav $\{v_1v_2, v_2v_3, v_3v_4, v_3v_5\}$). Bolj podrobno, rezultate iz tega poglavlja lahko povzamemo, kot sledi. Podali smo dve prevedbi problema: eno, ki temelji na nesoseščinah (izrek 9.4.3) in drugo, ki temelji na modularni dekompoziciji, v kombinaciji z Gausovo eliminacijo (izrek 9.2.7). Razvili smo polinomski algoritem za izračun sistema dobrega pokritja grafa brez vilic (izrek 9.5.5), kar posploši rezultat od Levita in Tankusa, kjer je problem rešen za grafe brez krempljev. Naš pristop implicira polinomsko prepoznavanje dobro-pokritih grafov brez vilic in posploši določene znane rezultate na kografih (izrek 9.3.1).

Poštena razdelitev s konfliktnim grafom. Razdelitev dobrin več agentom na zadovoljiv način je klasičen problem na področju kombinatorične optimizacije, kjer ima vsak agent svojo lastno aditivno funkcijo nad množico predmetov, cilj pa je dodeliti vsak predmet točno enemu od agentov, tako da je najmanj zadovoljen agent čim bolj zadovoljen (glej npr. [44,227]). Po navadi so problemi razdelitve opremljeni z nekaterimi dodatnimi omejitvami za dopustno razdelitev, obstajajo pa tudi različni modeli preferenc, ki jih lahko imajo agenti, in različne kriterijske funkcije, ki izhajajo iz teh. Klasični problem POŠTENA k-RAZDELITEV NEDELJIVIH PREDMETOV kot vhod prejme množico V sestavljeno iz n elementov, in k profitnih funkcij $p_1, \ldots, p_k : V \to \mathbb{Z}$ in izračuna tako k-particijo točk V, ki maksimizira minimalno zadovoljstvo vseh agentov.

V doktorski disertaciji smo študirali pošteno razdelitev n nedeljivih dobrin ali predmetov na množico k agentov s stališča teorije grafov in smo ji dodali nov vidik, tako da med pari predmetov dopuščamo nekompatibilnosti, ki so opisane s pomočjo konfliktnega grafa. To lahko odraža dejstvo, da predmeti izključujejo njihovo skupno uporabo, ali preprosto dejstvo, da so nekateri predmeti enake (ali podobne) vrste in ni smiselno, da en agent prejme več kot enega izmed teh predmetov. Takšna relacija je predstavljena s *konfliktnim grafom* $G = (V, E)$, kjer je V množica predmetov, povezave pa predstavljajo nekompatibilnosti med pari predmetov. Če sta dva elementa i in j povezana s povezavo $ij \in E$, potem i in j ne smeta biti vključena v isto podmnožico particije. Jasno je, da vsaka podmnožica elementov, dodeljenih enemu agentu, tvori neodvisno množico v tem grafu, v splošnem pa dovolimo delno razdelitev elementov, kar pomeni, da se lahko zgodi, da določeni elementi ostanejo nerazdeljeni. Takšna dodelitev predmetov agentom ustreza delnemu barvanju konfliktnega grafa, kjer je *delno*

k-barvanje grafa G zaporedje $(X_1, \ldots, X_k)$ k paroma disjunktnih neodvisnih množic grafa G, glej [22, 90]. Nekaj svežih rezultatov s tega področja, ki vsebujejo številne napotke na literaturo in preučujejo vprašanja poštene razdelitve, tudi v povezavi s strukturo grafov v ozadju problema, je dostopnih v [16,43]. *Raven zadovoljstva* delnega k-barvanja $(X_1, \ldots, X_k)$ grafa G (glede na profitne funkcije $p_1, \ldots, p_k$) je definirana kot minimum vseh rezultirajočih profitov $p_j(X_j) := \sum_{v \in X_j} p_j(v)$, kjer $j \in \{1, \ldots, k\}$. Vse to nas pripelje do študije naslednjega problema.

Poštena k-razdelitev s konflikti
Input: Graf $G = (V, E)$, k profitnih funkcij $p_1, \ldots, p_k : V \to \mathbb{Z}_+$.
Task: Poišči delno k-barvanje grafa G z maksimalno ravnijo zadovoljstva.

Z nekaj truda lahko vidimo, da je, tudi brez upoštevanja konfliktov, problem Poštene k-razdelitve nedeljivih predmetov šibko NP-težek za vsako konstanto $k \geq 2$ in krepko NP-težek, če je k del vhodnih podatkov. To velja tudi za k identičnih profitnih funkcij. Tako je obstoj psevdopolinomskih algoritmov za Pošteno k-razdelitev s konflikti možen le za konstanten k (razen če velja P=NP).

Opazimo, da v primeru, ko je $k = 1$, problem sovpada s problemom Najtežje neodvisne množice: v podanem grafu $G = (V, E)$ z utežno funkcijo na točkah grafa G poišči najtežjo neodvisno množico. V primeru, ko imamo enotne uteži in je $k = 1$, dobimo posplošitev problema Neodvisne množice, in zato sklepamo, da je Poštena 1-razdelitev s konflikti krepko NP-težek problem.

V doktorski disertaciji smo obravnavali zahtevnost problema Poštene k-razdelitve s konflikti za različne razrede konfliktnih grafov. Študirali smo mejo med krepko NP-težkimi primeri in tistimi, kjer lahko za vsako konstanto k zagotovimo obstoj psevdo-polinomskega algoritma.

Najprej pokažemo, da je za vse $k \geq 1$ pod določenimi pogoji odločitvena verzija problema Poštene k-razdelitve s konflikti krepko NP-polna za grafe konfliktov iz razreda $\mathcal{G}$, v katerem je problem Neodvisna množica NP-poln (glej poglavje 10.2).

Ko je graf konfliktov dvodelen, ali povezavni graf dvodelnega grafa, pokažemo, da je problem Poštene k-razdelitve s konflikti krepko NP-težek (izreka 10.3.1 in 10.3.2). Zanimivo je, da v omenjenih razredih grafov problem Neodvisne množice lahko rešimo v polinomskem času (glej [162, 213, 216]).

Po drugi strani pa, v primeru ko konfliktni graf pripada razredu bikonveksnih dvodelnih grafov, pokažemo, da je problem Poštene k-razdelitve s konflikti mogoče rešiti v psevdopolinomskem času (glej izrek 10.4.4). Ta rezultat temelji na psevdopolinomskem algoritmu za isti problem v primeru, ko konfliktni graf pripada razredu neprimerljivostnih grafov. Poleg naštetih rezultatov predstavimo polinomske algoritme, ki temeljijo na dinamičnem programiranju in rešijo problem za primer ko konfliktni graf pripada enem izmed naslednjih razredov grafov: tetivni grafi (izrek 10.4.12), grafi omejene drevesne širine (izrek 10.4.13), in grafi omejene klične širine (izrek 10.4.14).

Declaration

I declare that this thesis does not contain any material previously published or written by another person except where due reference is made in the text.

Nevena Pivač

Nevena Pivač