THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale Information Structures Systèmes

Unité de recherche UMR5506 (LIRMM)

Algorithms for graph modification problems: towards generality and efficiency

Algorithmes pour des problèmes de modification de graphes : Généralisation et efficacité

Présentée par Laure Morelle
le 23 septembre 2025

Sous la direction de Ignasi SAU VALLS et Dimitrios M. THILIKOS

Devant le jury composé de

Frédéric HAVET, DR., Université Côte d'Azur, CNRS, Inria, I3S	**Président du jury**
Robert GANIAN, A. Pr., Vienna University of Technology	**Rapporteur**
Eunjung KIM, A. Pr., Korea Advanced Institute of Science & Technology	**Rapporteuse**
Archontia GIANNOPOULOU, A. Pr., National & Kapodistrian University of Athens	**Examinatrice**
Petr GOLOVACH, Pr., University of Bergen	**Examinateur**
Ignasi SAU VALLS, DR., LIRMM, Université de Montpellier, CNRS	**Co-directeur de thèse**
Dimitrios M. THILIKOS, DR., LIRMM, Université de Montpellier, CNRS	**Co-directeur de thèse**

UNIVERSITÉ DE MONTPELLIER

Remerciements

I cannot start my acknowledgments by anyone else than my two supervisors, Ignasi Sau and Dimitrios Thilikos. Thank you Dimitrios for those great and enlightening conversation about research and theory making, and for always encouraging me to travel and visit researchers. Thank you Ignasi for your always careful reading of my writings, and for your great support me whenever I wished to travel. It is often said that a good PhD starts with good PhD supervisors, and I think it would be difficult to find better supervisors than the two of you.

I would also like to thank the members of my defense committee: Robert Ganian, Archontia Giannopoulou, Petr Golovach, Frédéric Havet, and Eunjung Kim, with a special mention to Robert Ganian and Eunjung Kim for accepting to review this thesis, and to Frédéric Havet for accepting to preside this committee. I am deeply grateful for your time and your effort.

I wish to express my thanks to all my coauthors: Ignasi and Dimitrios, of course, but also Davi de Andrade, Júlio Araújo, Matthias Bentert, Gaétan Berthe, Marthe Bonamy, Nicolas Bousquet, Quentin Chuet, Yoann Coudert–Osmont, Alexander Dobler, Lars Jaffke, Victor Falgas–Ravry, Fedor Fomin, Petr Golovach, Amaury Jacques, Timothé Picavet, Evangelos Protopapas, Amadeus Reinald, Mathis Rocton, Alexander Scott, Ana Silva, Giannos Stamoulis, and Sebastian Wiederrecht.

Mais une thèse ne se réduit pas à écrire des papiers : elle est construite sur des rencontres. Je voudrais remercier Chien-Chung Huang, qui m'a fait découvrir le monde de la recherche, ainsi que Pierre Aboulker, qui m'a introduit à la complexité paramétrée et qui m'a orienté vers Montpellier pour la thèse.

A deep thank to Petr Golovach and Fedor Fomin, who invited me twice to Bergen. I love the country, the city, the mountains, the team. Thank you both, as well as Matthias Bentert, for your time and the great research discussions.

Thank you Sebastian Wiederrecht and Sang-il Oum for hosting and supporting my visit to the IBS in South Korea. I enjoyed my three weeks there with all the team members. Thank you Sebastian in particular, for those fruitful discussions, that finally put an end to our project.

Merci Marthe pour m'avoir invité à Bordeaux. J'y ai passé un excellent moment, à travailler avec toi et Timothé, ainsi qu'à rencontrer les membres de l'équipe.

Je voudrais aussi bien évidemment remercier les membres de l'équipe AlGCo. Merci à Fanny, Vagelis, Giannos et Sebastian, qui m'ont accuelli à mon arrivée dans l'équipe. Merci à mes camarades de thèse, Gaétan, Amadeus et Yann, ainsi que ceux qui ont suivi (thésards et postdocs), Hugo, Simon, Raul, Guilherme, et à tous les stagiaires. On a passé de grands moments ensemble. Merci aussi à tous les membres permanents de l'équipe pour avoir toujours su garder l'ambiance conviviale et chaleureuse: William, Daniel, Petru, Marin, Mathieu, Stéphane, Mickaël, Christophe, Alexandre, Emeric, Pascal (and Matthieu). Merci aussi à l'équipe administrative, et notamment Nicolas Serrurier-Gourvès, best secretaire ever: tu nous manques beaucoup. Merci à toutes les autres que j'ai pu rencontrer au LIRMM ou à l'Université de Montpellier, et que je ne peux pas tous citer.

Thank you to all the people I met in the following workshops, conferences, and meetings: APGA 2022, ICGT 2022, GROW 2022, CoA 2022, JGA 2022, LoGAlg 2022, FPT Fest 2023, ICALP 2023, ALGO 2023, STRUG Bootcamp 2023, JGA 2023, JCALM 2023, Algoridam workshop, and JGA 2024. C'est notamment super de rencontrer la communauté française des graphes chaque année aux JGA.

Merci également à mes amis de longue date avec qui j'ai gardé le contact malgré la distance, aux nouveaux amis que j'ai pu me faire à Montpellier autour de jeux de société, et à toutes les personnes formidables que j'y ai croisées.

Merci à tous ceux que j'ai oubliés de mentionner et que je regretterai plus tard.

Et bien sûr pour finir, merci à ma famille, mes grands-parents, mes parents, ma sœur.

Abstract

In this thesis, we study graph modification problems, where the goal is to do the minimum amount of modifications of some kind to an input graph so that the modified graph belongs to some target graph class $\mathcal{H}$. Many well-known problems on graphs can be expressed as graph modification problems, hence the considerable attention given to this subject.

More particularly, we study graph modification problems under the framework of parameterized complexity, which focuses on expressing the running time of an algorithm not only in terms of the size of the input n, but also in terms of other parameters. The parameter that we consider here is usually the size k of the modulator (i.e., the modified vertices of the input graph), or another, more refined, measure on the modulator. More particularly, we look for FPT-algorithms, that is, algorithms running in time $f(k) \cdot n^c$ for some function f and some constant c.

Graph modification problems have already been extensively studied. In particular, there exist algorithmic meta-theorems solving a considerable number of graph modification problems in time $f(k) \cdot n^2$ (for some huge function f) when the target graph class $\mathcal{H}$ is minor-closed and the so-called bidimensionality of the modulator is bounded by k.

The results of this thesis are divided into three parts.

In the first part, we give an example of the use of bounded bidimensionality modulators via a structure theorem: we prove that a graph that excludes an edge-apex graph (a graph that is planar after removing an edge) as a minor is a clique-sum of graphs that can be embedded in the projective plane after doing identifications in a set of vertices of bounded bidimensionality.

In the second part, we design FPT-algorithms with explicit and moderate parametric dependencies for many graph modification problems that are already solvable using algorithmic meta-theorems, in particular improving the dependence on k. We study most particularly problems where the target class is minor-closed, the measure on the modulator is its size or the treedepth of its torso, and the modification operation is the identification of vertices (which is the modification operation of the aforementioned structure theorem), the deletion of vertices (which is the most studied modification operation), or even any combination of vertex identifications/deletions and edge additions/deletions.

In the third and last part, we develop new techniques to solve graph modification problems beyond the scope of current algorithmic meta-theorems. As such, we provide a general dynamic programming scheme solving problems parameterized by bipartite treewidth, a parameter generalizing both the treewidth and the odd cycle transversal number and that is closely related to odd-minor-closed graph classes (which are more general than minor-closed graph classes). Furthermore, we design a new irrelevant vertex technique that we use to solve graph modification problems beyond the limit of bidimensionality.

Keywords: Graph Modification Problems, Parameterized Complexity, Graph Minor Theory, Structure Theorems.

Résumé

Dans cette thèse, nous étudions des problèmes de modification de graphes, dont le but est d'appliquer un minimum de modifications d'un certain type au graphe donné en entrée, de sorte que le graphe résultant appartienne à une certaine classe de graphes $\mathcal{H}$ cible. Beaucoup de problèmes connus peuvent être exprimés comme des problèmes de modification de graphes, d'où l'intérêt considérable porté au sujet.

Plus précisément, nous étudions des problèmes de modification de graphes sous le point de vue de la complexité paramétrée, qui consiste à exprimer le temps d'exécution d'un algorithme non seulement en fonction de la taille n de l'entrée, mais aussi en fonction d'autres paramètres. Le paramètre considéré ici est généralement la taille k du modulateur (i.e., les sommets du graphe donné en entrée qui sont modifiés), ou une autre mesure sur le modulateur. Nous cherchons principalement des algorithmes dit FPT, c'est-à-dire qui terminent en temps $f(k) \cdot n^c$ pour une fonction f et une constante c.

Beaucoup de résultats existent déjà concernant les problèmes de modification de graphes. En particulier, il existe des méta-théorèmes algorithmiques qui résolvent un nombre considérable de problèmes de modification de graphes en temps $f(k) \cdot n^2$ (où f est une fonction croissant très rapidement) quand la classe cible $\mathcal{H}$ est close par mineur et un paramètre nommé bidimensionnalité est borné par k pour le modulateur.

Les résultats de la thèse sont divisés en trois parties.

Dans une première partie, nous prouvons un théorème de structure qui peut s'énoncer en terme d'identification de sommets dans un ensemble de bidimensionnalité bornée.

Dans une seconde partie, nous construisons des algorithmes FPT avec une meilleure complexité (notamment concernant la dépendance en k) pour de nombreux problèmes de modification de graphes qui sont déjà résolubles par des méta-théorèmes algorithmiques. Nous étudions des problèmes où la classe cible $\mathcal{H}$ est close par mineur, la mesure sur le modulateur est sa taille ou la profondeur arborescente de son torse, et l'opération de modification est l'identification de sommets (comme pour le théorème de structure), la suppression de sommets, ou même parfois n'importe quelle combinaison de suppression/identification de sommets et de suppression/addition d'arêtes.

Dans une troisième partie, nous développons de nouvelles techniques pour résoudre des problèmes de modification au-delà des méta-théorèmes algorithmiques existant déjà. Ainsi, nous proposons une méthode de programmation dynamique pour résoudre des problèmes paramétrés par la largeur arborescente bipartie, un paramètre fortement relié aux classes de graphes closes par mineur impair (qui sont plus générales que les classes de graphes closes par mineur). En outre, nous créons une nouvelle technique du sommet inutile que nous utilisons pour résoudre des problèmes de modification de graphes où le modulateur n'a pas forcément une bidimensionnalité bornée.

Mot-clés : Problèmes de modification de graphes, Complexité paramétrée, Mineurs de graphes, Théorèmes de structure.

Résumé étendu en français

Pour votre anniversaire, imaginez que vous voulez inviter un maximum d'amis. Cependant, certains de vos amis ne s'entendent pas entre eux et se disputeront dès qu'ils se rencontreront. C'est pourquoi vous voulez choisir un minimum de personnes à ne *pas* inviter de sorte qu'il n'y ait aucune dispute.

Cette situation peut être modélisée par un graphe. Un *graphe* G est une structure mathématique composée d'un ensemble $V(G)$ de *sommets* et d'un ensemble $E(G)$ de paires $e = \{u, v\}$ de sommets qu'on appelle des *arêtes* (cf. Figure 1).

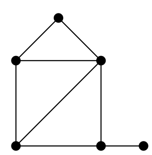

FIGURE 1 : Un graphe avec six sommets et huit arêtes.

Dans le cas qui nous intéresse ici, les sommets du graphe sont vos amis, et deux sommets sont reliés s'il y a un conflit entre les deux personnes correspondantes. Modélisé par un graphe, le problème que nous cherchons à résoudre est le problème de COUVERTURE PAR SOMMETS. Étant donné un graphe G, l'objectif est de trouver le plus petit ensemble S de sommets de G tel que le graphe $G - S$ obtenu après avoir supprimé les sommets dans S n'ait pas d'arêtes. Un tel ensemble S s'appelle une *couverture par sommets*.

Étant donné que vous avez beaucoup d'anniversaires à venir, vous souhaitez créer un ensemble d'instructions à donner à un ordinateur de sorte que celui-ci, quand il reçoit un graphe G, retourne une couverture par sommets de taille minimum. Un tel ensemble d'instructions est ce que l'on appelle un *algorithme*.

Un point critique dans un algorithme est sa *complexité temporelle*, c'est-à-dire le temps qu'il lui faut pour retourner une solution dans le pire des cas. L'objectif est bien sûr de trouver une solution aussi rapidement que possible.

Pour le problème de COUVERTURE PAR SOMMETS, l'algorithme le plus simple consisterait à tester, pour chaque sous-ensemble $S \subseteq V(G)$ de sommets de G, si c'est une couverture par sommets, et de retourner une couverture par sommets de taille minimum. Si n est le nombre de sommets de G, alors le nombre de sous-ensembles de sommets est 2^n. En conséquence, cet algorithme a une complexité en temps de $\mathcal{O}(2^n)$: on parle d'un algorithme *exponentiel*. Le problème ici est que votre nombre d'amis va certainement augmenter avec le temps. Si vous avez un jour cent amis, votre anniversaire sera terminé le temps que l'ordinateur renvoie une solution. Voici pourquoi nous recherchons plutôt des algorithmes avec une complexité temporelle de $\mathcal{O}(n^c)$ pour une constante c (un algorithme *polynomial*), ou même $\mathcal{O}(n)$ (un algorithme *linéaire*).

Malheureusement, COUVERTURE PAR SOMMETS est ce que l'on appelle un problème NP-dur [181], ce qui signifie qu'il est peu probable qu'il existe un algorithme polynomial le résolvant. Il existe plusieurs méthodes permettant de contourner cet obstacle. Celle que nous utilisons tout particulièrement ici est la *complexité paramétrée*. L'idée est que la difficulté du problème ne vient peut-être pas du nombre de sommets n du graphe G donné en entrée, mais plutôt d'un paramètre k plus fin, qui est une mesure sur G ou sur la solution. Nous pouvons espérer que k est petit dans notre cas, ce qui nous permettrait de chercher un algorithme dont la complexité temporelle est polynomiale en n après avoir fixé k. Nous cherchons plus particulièrement des algorithmes terminant en temps $f(k) \cdot n^c$, où f est une fonction et c est une constante. Un tel algorithme est ce qu'on appelle un *algorithme* FPT.

Dans notre problème d'anniversaire, nous ne voulons pas refuser trop de monde. Peut-être que s'il faut refuser $k \geq 4$ personnes pour éviter toute dispute, nous abandonnerons et inviterons tout le monde, même s'il y aura des conflits. Cela correspond à fixer la taille k d'une couverture par sommet minimum : l'algorithme retourne soit une couverture par sommet minimum de taille au plus k, soit déclare qu'une telle couverture n'existe pas. Pour COUVERTURE PAR SOMMETS, il existe de tels algorithmes paramétrés par la taille k de la solution. Le meilleur qui a été trouvé jusqu'à présent a une complexité temporelle de $\mathcal{O}(1.2738^k + kn)$ [59].

COUVERTURE PAR SOMMETS est ce qu'on appelle un *problème de modification de graphes*. Chaque problème de modification de graphes est caractérisé par :

- une classe de graphes *cible* $\mathcal{H}$,
- un ensemble de *modifications* autorisées $\mathcal{M}$, et
- une *mesure* p sur le *modulateur* X du graphe G donné en entrée, c'est-à-dire le sous-ensemble de sommets qui sont soit modifiés, soit incidents à une arête modifiée.

Dans le cas du problème de COUVERTURE PAR SOMMETS, la classe cible est l'ensemble $\mathcal{H}$ des graphes sans arêtes, l'ensemble de modifications autorisées est $\mathcal{M} = \{\text{suppression de sommets}\}$, et la mesure p sur le modulateur X (qui est ici l'ensemble des sommets supprimés) est sa taille $|X|$.

Étant donné que chaque choix de $\mathcal{H}$, $\mathcal{M}$, et p crée un nouveau problème de modification de graphes, le nombre de ceux-ci est infini. C'est pourquoi deux lignes de recherches principales ont émergé concernant les problèmes de modification de graphes :

- **Efficacité :** Concernant les problèmes les plus connus, à commencer par COUVERTURE PAR SOMMETS, les chercheurs tentent d'optimiser au maximum la complexité des algorithmes les résolvant.
- **Généralité :** Plutôt que de résoudre chaque problème un par un, notamment ceux dont l'utilité reste très anecdotique, les chercheurs prouvent des méta-théorèmes algorithmiques permettant de construire des algorithmes pour un vaste ensemble de problèmes simultanément, au prix d'un manque d'optimisation de leur complexité (notamment leur dépendance en k).

Voyons différentes méthodes permettant d'englober de nombreux problèmes de modification de graphes dans un même théorème.

Classes de graphes cibles

Certaines classes de graphes cibles $\mathcal{H}$ ont des propriétés intéressantes qui rendent plus facile la résolution de problèmes de modification vers $\mathcal{H}$. C'est le cas de la propriété d'être clos par mineur.

Un graphe H est un *mineur* d'un graphe G s'il peut être obtenu en supprimant des arêtes et des sommets de G, et en contractant des arêtes de G (cf. Figure 2). Une classe de graphes $\mathcal{H}$ est *close par mineur* si, pour tout graphe $G \in \mathcal{H}$, ses mineurs sont aussi dans $\mathcal{H}$.

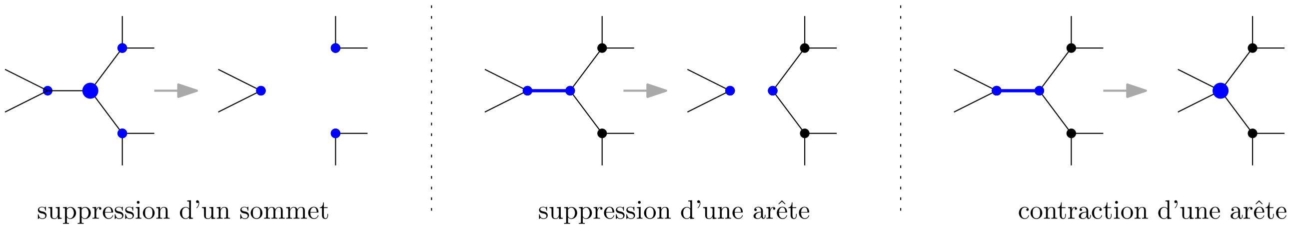

FIGURE 2 : Illustration d'une suppression d'un sommet, d'une suppression d'une arête et d'une contraction d'une arête.

Un résultat important sur les classes de graphes closes par mineur, prouvé par Robertson et Seymour [278] dans leur célèbre série de 23 papiers sur la théorie des mineurs, est qu'elles ont un nombre fini d'*obstructions* (les graphes qui n'appartiennent pas à la classe de graphes, mais dont tous les mineurs font partie). Cela signifie que pour tester si un graphe G appartient à une classe de graphes $\mathcal{H}$ close par mineur, il suffit de tester si chacune des obstructions F de $\mathcal{H}$ est un mineur de G. Étant donné que ce test peut s'effectuer en temps $f(|V(F)|) \cdot n^{1+o(1)}$ [205], il est donc possible de tester si $G \in \mathcal{H}$ en temps $f(s_{\mathcal{H}}) \cdot n^{1+o(1)}$, où $s_{\mathcal{H}}$ est la taille de la plus grande obstruction de $\mathcal{H}$.

Ce qui nous intéresse est la chose suivante : de nombreux problèmes de modification de graphes peuvent se réduire à tester si le graphe donné en entrée appartient à une certaine classe de graphes qui est close par mineur. Par exemple, appelons SUPPRESSION DE SOMMETS VERS $\mathcal{H}$ le problème de modification de graphes qui demande, étant donné un graphe G et un entier k, si on peut supprimer au plus k sommets de G de sorte que le graphe obtenu appartienne à $\mathcal{H}$, et supposons que $\mathcal{H}$ est une classe close par mineur. Pour k fixé, l'ensemble $\mathcal{A}_k(\mathcal{H})$ des graphes G tels que (G, k) est une instance positive du problème est une classe de graphes close par mineur. Par conséquent, pour résoudre SUPPRESSION DE SOMMETS VERS $\mathcal{H}$ pour une instance (G, k), il suffit de tester si G appartient à $\mathcal{A}_k(\mathcal{H})$, ce qui, nous l'avons vu, est faisable en temps quasi-linéaire paramétré par k et $s_{\mathcal{H}}$.

Malheureusement, ce résultat ne donne aucune garantie sur la dépendance en k. Pire, le résultat de Robertson et Seymour est non-constructif, c'est-à-dire qu'il donne l'existence d'un nombre fini d'obstructions, mais n'explique pas comment les construire. En conséquence, cela nous donne l'existence d'algorithmes pour certains problèmes de modification de graphes, mais pas leur construction. D'autres chercheurs ont néanmoins prouvé comment construire ces obstructions dans certain cas. En particulier, nous savons comment construire les obstructions de $\mathcal{A}_k(\mathcal{H})$ quand les obstructions de $\mathcal{H}$ sont connues [160, 285]. Depuis, des algorithmes avec une meilleure dépendance ont été proposés, qui s'appuient généralement sur les techniques développées par Robertson et Seymour, notamment la "technique du sommet inutile" [271], sur laquelle nous reviendrons plus tard. L'algorithme avec la meilleure dépendance en k pour SUPPRESSION DE SOMMETS VERS $\mathcal{H}$ avec $\mathcal{H}$ clos par mineur a actuellement une complexité $2^{\mathsf{poly}(k)} \cdot n^3$, où poly est une fonction polynomiale dont le degré dépend de $s_{\mathcal{H}}$ [284].

Une autre propriété, plus générale que la clôture par mineur, pourrait impliquer des résultats similaires. Il s'agit de la clôture par mineur impair.

Un *mineur impair* H de G est essentiellement un mineur de G, mais qui préserve la parité des cycles. C'est-à-dire que la taille d'un cycle dans H à la même parité que celle du cycle correspondant dans G. Une classe de graphes close par mineur impair est aussi close par mineur, mais le contraire n'est pas vrai, d'où le fait que la clôture par mineur impair est plus générale que la clôture par mineur.

Il a été affirmé (sans preuve) qu'une classe qui est close par mineur impair a un nombre fini d'obstructions (en tant que mineur impair) [168]. Additionnellement, des algorithmes existent pour tester si un graphe F est un mineur impair d'un graphe G [168, 193]. En conséquence, les résultats sur les classes closes par mineur devraient pouvoir être étendus aux classes closes par mineur impair. Malheureusement, la théorie sur les mineurs impairs est bien moins développée que celle sur les mineurs, et beaucoup de résultats restent à prouver sur le sujet.

Modifications

Après les classes cibles, parlons des différents types de modifications.

Outre la *suppression de sommets*, les opérations de modification les plus étudiées sont certainement la *suppression d'arêtes*, l'*ajout d'arêtes*, et la *contraction d'arêtes*. Nous pouvons ajouter à cela d'autres opérations plus exotiques, telles que la *suppression d'un ensemble connexe de sommets*, la *suppression d'un ensemble indépendant*, la *suppression* ou la *contraction d'un couplage*, ou encore la *complémentation d'un sous-graphe*. Il est aussi possible de combiner plusieurs opérations de modification ensemble. Par exemple, si on autorise à la fois la suppression et l'addition d'arêtes, on parle d'*édition d'arêtes*.

Une méthode pour englober plusieurs types de modification dans une unique structure est la suivante. Une *action de remplacement* est une fonction $\mathcal{L}$ qui associe chaque graphe H_1 à un ensemble

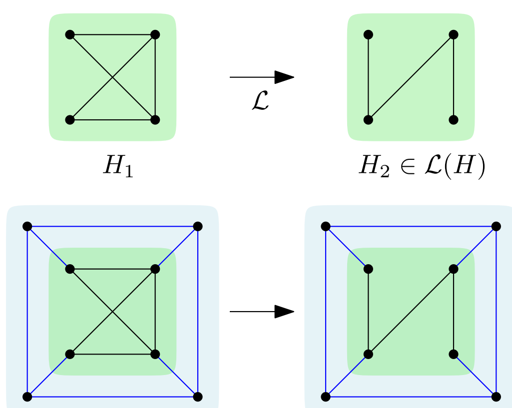

FIGURE 3 : Illustration d'une modification de graphes autorisée par l'action de remplacement $\mathcal{L}$.

$\mathcal{L}(H_1)$ de graphes H_2 tels que $|V(H_1)| = |V(H_2)|$ (cf. Figure 3). Le problème de $\mathcal{L}$-REMPLACEMENT VERS $\mathcal{H}$ demande alors, étant donné un graphe G et un entier k, s'il y a un sous-graphe H_1 de G et un graphe H_2 dans $\mathcal{L}(H_1)$ de sorte que le graphe obtenu en remplaçant H_1 par H_2 dans G appartienne à $\mathcal{H}$. Intuitivement, H_1 correspond au modulateur du graphe G en entrée de notre problème de modification de graphes, et H_2 correspond à l'ensemble des modifications faites sur H_1. Par exemple, si notre problème demande s'il est possible d'enlever au plus k arêtes à G pour appartenir à $\mathcal{H}$, alors $\mathcal{L}(H_1)$ est l'ensemble des graphes pouvant être obtenus depuis H_1 en enlevant au plus k arêtes. $\mathcal{L}$-REMPLACEMENT VERS $\mathcal{H}$ peut simuler tout problème de modification de graphes où la mesure sur le modulateur est sa taille et la modification est n'importe quelle combinaison d'ajouts et de suppressions d'arêtes. Il a été prouvé que $\mathcal{L}$-Remplacement vers $\mathcal{H}$ peut être résolu en temps $\mathcal{O}(f(k)\cdot n^2)$ quand $\mathcal{H}$ est la classe des graphes planaires, c'est-à-dire des graphes pouvant être dessinés dans un plan sans croisement d'arêtes [121].

Mesures sur le modulateur

Concernant les différentes mesures sur le modulateur, les chercheurs s'intéressent dans la plupart des cas à sa taille, comme vu dans les différents problèmes présentés ci-dessus. Cependant, d'autres mesures existent.

Commençons par quelques définitions. Le *torse* d'un ensemble X de sommets dans un graphe G, noté $\mathsf{torso}(G,X)$, est le graphe obtenu en supprimant chaque composante C de $G-X$ et en rajoutant, si elle n'existe pas déjà, une arête entre chaque paire de sommets de X adjacent à un sommet de C (cf. Figure 4). Étant donné une classe de graphes $\mathcal{H}$ et un paramètre p sur les graphes,

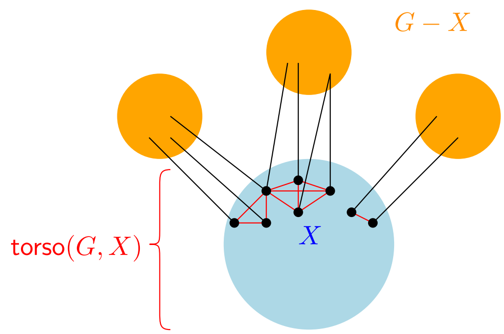

Figure 4 : Torse d'un ensemble X dans un graphe G. For chaque composante connexe C de $G-X$ (en orange), les arêtes (en rouge) entre chaque paire de sommets de X adjacente à un sommet de C sont ajoutées, si elles n'existent pas déjà.

le paramètre $\mathcal{H}$-p est le paramètre défini par, pour tout graphe G,

$$\mathcal{H}\text{-}\mathsf{p}(G)=\min\{k\in\mathbb{N}\mid \exists X\subseteq V(G), \mathsf{p}(\mathsf{torso}(G,X))\leq k \text{ et les composantes de } G-X \text{ sont dans } \mathcal{H}\}.$$

Évidemment, si p est le paramètre size qui affecte à chaque graphe son nombre de sommets et que $\mathcal{H}$ est clos par union disjointe, alors le problème de Suppression de Sommets vers $\mathcal{H}$ correspond à, étant donné un graphe G et un entier k, tester si $\mathcal{H}$-$\mathsf{p}(G)\leq k$.

Deux autres paramètres p ont été bien étudiés dans ce contexte. Le premier est la *profondeur arborescente* d'un graphe G, noté $\mathsf{td}(G)$. L'idée est de retirer un sommet de chaque composante connexe à chaque itération, et la profondeur arborescente de G est le nombre minimum d'itérations nécessaire pour supprimer tous les sommets. Le paramètre $\mathcal{H}$-td est souvent nommé *distance d'élimination à* $\mathcal{H}$ [43,44], et peut être reformulé de la façon suivante : un sommet est retiré de chaque composante connexe à chaque itération, et la distance d'élimination à $\mathcal{H}$ est le nombre minimum d'itérations nécessaire pour que chaque composante connexe appartiennent à $\mathcal{H}$. Le problème correspondant, à savoir tester, étant donné une instance (G,k), si $\mathcal{H}$-$\mathsf{td}(G)\leq k$, est nommé Distance d'Élimination à $\mathcal{H}$.

Le second paramètre est la *largeur arborescente* d'un graphe G, noté $\mathsf{tw}(G)$. L'idée est de mesurer à quel point G ressemble à une forêt, c'est-à-dire un graphe sans cycle. Ce paramètre, que nous ne définirons pas formellement dans ce résumé, est très apprécié par les chercheurs car de nombreux problèmes admettent un algorithme FPT quand le paramètre est la largeur arborescente du graphe donné en entrée [18,67]. Le paramètre $\mathcal{H}$-tw a été développé dans [104] pour combiner les propriétés intéressantes de la largeur arborescente et de la cible $\mathcal{H}$.

Un résultat important sur ces différentes mesures sur le modulateur a été prouvé par Agrawal, Kanesh, Lokshtanov, Panolan, Ramanujan, Saurabh, et Zehavi dans [6] : pour tout $\mathsf{p} \in \{\mathsf{size}, \mathsf{td}, \mathsf{tw}\}$ et pour toute classe cible $\mathcal{H}$ ayant les “bonnes” propriétés (qui sont peu contraignantes), si le problème de tester si $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ admet un algorithme FPT, alors la même chose est vraie pour les deux autres paramètres. Plus généralement, leur résultat semble pouvoir s'appliquer à tout paramètre p *plus grand* que tw (i.e. tel qu'il existe une fonction $f : \mathbb{N} \to \mathbb{N}$ telle que, pour tout graphe G, $\mathsf{tw}(G) \leq f(\mathsf{p}(G))$), dont font partie size et td.

Un méta-théorème algorithmique

Sau, Stamoulis, et Thilikos ont prouvé dans [287] un théorème englobant tous les problèmes vus ci-dessus, tant que la classe cible $\mathcal{H}$ est close par mineur, et qui construit pour chacun de ces problèmes un algorithme FPT de complexité $f(k) \cdot n^2$ pour une certaine fonction f. C'est ce qu'on appelle un *méta-théorème algorithmique*.

Ce méta-théorème emploie une mesure sur le modulateur qui n'utilise pas le torso. Elle s'appuie à la place sur les *paramètres annotés*. Étant donné un ensemble X de sommets d'un graphe G, un *X-mineur* de G est un mineur de G tel que chaque ensemble d'arêtes connectées qui est contracté contient un sommet de X. La version annotée d'un paramètre p sur les graphes est alors le paramètre qui, à chaque graphe G et à chaque ensemble $X \subseteq V(G)$, associe

$$\mathsf{p}(G, X) = \min\{k \in \mathbb{N} \mid \text{il existe un } X\text{-mineur } H \text{ de } G \text{ tel que } \mathsf{p}(H) \leq k\}.$$

En particulier, pour tout graphe G et tout $X \subseteq V(G)$, $\mathsf{tw}(G, X) \leq \mathsf{tw}(\mathsf{torso}(G, X))$. Le méta-théorème algorithmique de [287] s'applique aux problèmes de modification de graphes où la mesure sur le modulateur a largeur arborescente annotée bornée. Cela contient notamment le problème de tester si $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ pour tout paramètre p plus grand que tw.

Plutôt que de parler de largeur arborescente annotée bornée, on parle de façon équivalente de *bidimensionnalité* bornée. Les auteurs de [287] utilisent la technique du sommet inutile de Robertson et Seymour [271] mentionnée plus haut, et essaient de l'appliquer de la façon la plus générale possible. En l'occurrence, cette méthode ne fonctionne pas quand le modulateur a une bidimensionnalité non bornée, donc résoudre des problèmes où le modulateur a une bidimensionnalité non bornée requiert d'autre méthodes.

Quand bien même nous présentons plus haut un problème de modification où la mesure sur le modulateur X est la largeur arborescente du torse $\mathsf{tw}(\mathsf{torso}(G, X))$, nous ne connaissons pas de problème de modification de graphes étudié dans la littérature où la mesure est la bidimensionnalité du modulateur, ou, dit autrement, correspond à la limite du méta-théorème algorithmique de [287]. Cependant, des modulateurs de bidimensionnalité bornée sont présents dans d'autres types de résultats : les théorèmes de structure.

Théorêmes de structure

Un *théorème de structure* est essentiellement un théorème décrivant la structure des graphes ayant une certain propriété, qui correspond souvent à l'exclusion d'un graphe H d'une certaine manière.

En particulier, le *théorème de structure des mineurs* [275] dit (de manière très schématique) que tout graphe qui ne contient pas un certain graphe H en tant que *mineur* est obtenu en “collant” ensemble des graphes qui peuvent être dessinés dans une certaine surface sans que des arêtes se croisent, à quelques “erreurs” près dont le nombre dépend de $|V(H)|$ (voir Figure 5 pour une illustration). Ce théorème a été reformulé à l'aide de modulateurs de bidimensionnalité bornée dans [302] : tout graphe qui ne contient pas un certain graphe H en tant que mineur est obtenu

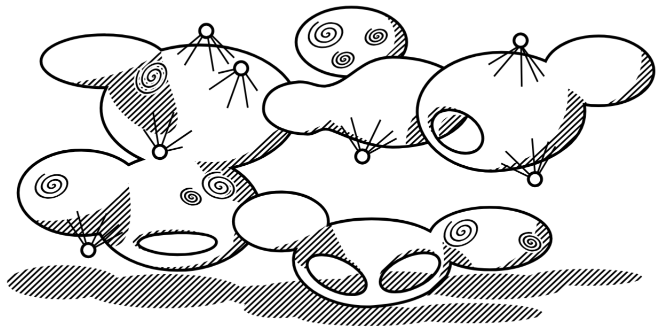

FIGURE 5 : Illustration artistique par Felix Reidl de la structure d'un graphe excluant un autre graphe en tant que mineur.

en "collant" ensemble des graphes qui peuvent être dessinés dans une certaine surface après avoir supprimé un ensemble X de sommets de bidimensionnalité bornée.

D'autre théorèmes de structure existent si H possède des propriétés plus restrictives, comme être planaire [261], ou planaire après avoir supprimé un sommet [101].

Il existe aussi un théorème de structure concernant les mineurs impairs [85] : tout graphe qui ne contient pas un certain graphe H en tant que *mineur impair* est obtenu en "collant" ensemble des graphes qui sont soit bipartis (i.e., sans cycle impair) soit excluent un graphe H' en tant que mineur, à quelques erreurs près dont le nombre dépend de $|V(H)|$. Ce résultat a ensuite été reformulé de la manière suivante en terme de modulateurs [299] : pour tout graphe H, il existe une fonction c_H telle que, pour tout graphe G excluant H en tant que mineur, $\mathcal{H}\text{-}\mathsf{btw}(G) \leq c_H$. Ici, btw dénote la *largeur arborescente bipartie*, qui est un paramètre généralisant à la fois la largeur arborescente et les graphes bipartis. Donc, pour résoudre des problèmes sur les classes de graphes closes par mineur impair, une première étape est de résoudre ces problèmes paramétrés par btw.

Organisation et résultats de cette thèse

Dans la Partie I, nous introduisons plus en détail le contexte et les motivations de cette thèse (Chapitre 1), ainsi que les résultats obtenus (Chapitre 2) et les principales techniques employées (Chapitre 3). Nous définissons aussi dans le Chapitre 4 les notions utilisées tout au long de la thèse.

Dans la Partie II (et plus précisément le Chapitre 5), nous présentons un théorème de structure pour l'exclusion d'un graphe H en tant que mineur. Ici, le graphe H considéré appartient à une autre classe de graphes proche de la planarité : on considère les graphes H qui sont planaires après avoir supprimé une arête. On appelle de tels graphes des graphes *arête-apex*. Le théorème de structure obtenu est donc plus fin que le théorème de structure des mineurs [275], et il peut notamment être reformulé en terme de modulateurs de bidimensionnalité bornée. Cependant, la modification n'est pas ici la suppression de sommets, mais l'*identification* de sommets (l'identification d'une paire $\{u, v\}$ de sommets est la même chose que la contraction de l'arête $\{u, v\}$, mais ici, on ne requiert pas la présence d'une arête). Ainsi, on prouve la propriété suivante : un graphe excluant un graphe arête-apex H en tant que mineur est obtenu en "collant" ensemble des graphes qui peuvent être dessinés dans une certaine surface après avoir effectué des identifications dans un ensemble X de sommets de bidimensionnalité bornée.

Dans la Partie III, nous obtenons des algorithmes avec une meilleure complexité en temps, notamment concernant la dépendance en k, pour une importante variété de problèmes de modifications de graphes qui sont déjà résolubles par le méta-théorème algorithmique de [287] en temps $f(k) \cdot n^2$ pour une fonction f non-explicitée.

Plus précisément, étant donné que nous requérons comme modification dans le Chapitre 5 l'identification de sommets, et que cette opération n'a encore jamais été étudiée, nous commençons l'exploration de ce type de modification dans le Chapitre 6. Nous étudions le problème de IDENTIFICATION VERS $\mathcal{H}$, c'est-à-dire le problème de modification de graphes où la classe cible est $\mathcal{H}$, la modification est l'identification de sommets, et la mesure sur le modulateur est sa taille. Dans le Chapitre 6, nous étudions l'une des classes $\mathcal{H}$ les plus simples non triviales, à savoir l'ensemble des forêts (les graphes sans cycles). En particulier, nous construisons un algorithme résolvant le problème en temps $\mathcal{O}(1.2738^k + k\sqrt{\log k} \cdot n)$ et nous proposons une façon de construire les obstructions (en tant que mineurs) de la classe $\mathcal{F}^{(k)}$ des graphes G tels que (G, k) est une instance positive de IDENTIFICATION VERS UNE FORÊT.

Dans le Chapitre 7, nous voulons étudier plus généralement IDENTIFICATION VERS $\mathcal{H}$ pour n'importe quelle classe $\mathcal{H}$ qui est close par mineur. Nous obtenons un algorithme de complexité $2^{\mathsf{poly}(k)} \cdot n^2$ (où poly est une fonction polynomiale dont le degré dépend de la taille maximale d'une obstruction de $\mathcal{H}$), qui résout non seulement IDENTIFICATION VERS $\mathcal{H}$, mais aussi SUPPRESSION DE SOMMETS VERS $\mathcal{H}$, ainsi qu'un nombre conséquent de problèmes de modification de graphes, pour $\mathcal{H}$ clos par mineur. Pour englober tous ces problèmes en un unique méta-théorème algorithmique, nous réadaptons le problème de $\mathcal{L}$-REMPLACEMENT VERS $\mathcal{H}$ de [121] pour qu'il puisse décrire des problèmes de modification de graphes où les modifications sont des combinaisons, non seulement d'additions et de suppressions d'arêtes comme dans [121], mais aussi de suppressions et d'identifications de sommets. Pour cela, nous redéfinissons la notion d'action de remplacement : si dans [121], une action de remplacement $\mathcal{L}$ associe chaque graphe à une collection de graphes de *même taille*, un graphe G peut maintenant être associé à une collection de graphes de *taille au plus celle de* G. Notre algorithme a ainsi une meilleure complexité que l'ancien meilleur algorithme résolvant SUPPRESSION DE SOMMETS VERS $\mathcal{H}$ en temps $2^{\mathsf{poly}(k)} \cdot n^3$ [284] et est le premier algorithme avec une dépendance *explicite* en k résolvant les autres problèmes de modifications de graphes qui peuvent être simulés par $\mathcal{L}$-REMPLACEMENT VERS $\mathcal{H}$. Le degré de la fonction poly est très grand, mais quand $\mathcal{H}$ est l'ensemble des graphes que l'on peut dessiner dans une certaine surface Σ, nous obtenons un meilleur algorithme avec une complexité $2^{\mathcal{O}(k^9)} \cdot n^2$.

Dans le Chapitre 8, nous changeons la mesure sur le modulateur : au lieu de la taille du modulateur, nous étudions la profondeur arborescente de son torse. Plus précisément, nous étudions le problème de DISTANCE D'ÉLIMINATION À $\mathcal{H}$ présenté plus haut. En utilisant des techniques similaires à celles employées dans le Chapitre 7, nous obtenons le premier algorithme FPT résolvant DISTANCE D'ÉLIMINATION À $\mathcal{H}$ pour n'importe quelle classe $\mathcal{H}$ qui est close par mineur avec une dépendance *explicite* en k. En outre, nous proposons une façon de construire les obstructions (en tant que mineurs) de la classe $\mathcal{E}_k(\mathcal{H})$ des graphes G tels que (G, k) est une instance positive de DISTANCE D'ÉLIMINATION À $\mathcal{H}$.

Dans la Partie IV, nous nous intéressons à des problèmes de modification de graphes qui ne sont pour l'instant résolubles par aucun méta-théorème algorithmique.

Comme présenté plus haut, plutôt qu'étudier des classes closes par mineur, on peut s'intéresser aux classes closes par *mineur impair*, et une première étape dans cette direction est la paramétrisation par la largeur arborescente bipartie btw. Dans le Chapitre 9, nous proposons une technique pour obtenir des algorithmes FPT paramétrés par btw, et nous appliquons cette technique pour résoudre un certain nombre de problèmes de modification de graphes paramétrés par btw.

Enfin, étant donné que la plupart des méta-théorèmes algorithmiques concernant des problèmes de modification de graphes, à commencer par celui de [287], utilise la technique du sommet inutile de Robertson et Seymour [271], qui ne s'applique que quand le modulateur a une bidimensionnalité bornée, nous créons dans le Chapitre 10 une nouvelle technique de sommet inutile, qui elle, fonctionne pour des modulateurs de bidimensionnalité non-bornée. Nous appliquons cette technique pour résoudre en temps polynomial le problème de $\mathcal{H}$-PLANARITÉ qui demande, étant donné un graphe G, s'il existe un ensemble X dont le torse est planaire et tel que les composantes connexes de $G - X$ sont dans $\mathcal{H}$, avec $\mathcal{H}$ obéissant à des propriétés moins restrictives que la clôture par mineur. Nous appliquons aussi cette technique pour résoudre en temps FPT le problème qui demande, étant donné un graphe G et un entier k, si $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$, pour deux paramètres p *plus petits* que tw, à savoir la *profondeur arborescente planaire*, qui combine td avec la planarité, et la *largeur arborescente planaire*, qui combine tw avec la planarité, étant donné, encore une fois, des conditions peu contraignantes sur $\mathcal{H}$. Dans ces trois cas, le modulateur, dont le torse est soit planaire, ou a une profondeur arborescente planaire bornée, ou a une largeur arborescente planaire bornée, peut avoir une bidimensionnalité non bornée, et c'est la première fois que de tels modulateurs sont étudiés pour des problèmes de modification de graphes.

Dans la Partie V, nous concluons la thèse avec un chapitre (Chapitre 11) proposant des questions ouvertes et des conjectures découlant des résultats de la thèse.

Contents

Part I

Introduction

CHAPTER 1

Motivation

Imagine that you want to invite your friends to your birthday party. However, you know that some of your friends do not like each other and will fight as soon as they meet. So your goal is to invite as many of your friends as possible, while ensuring that no fight erupts.

Graphs

This situation can be modeled by a graph [87]. A *graph* G is a mathematical structure composed of a set $V(G)$ of points, that we call *vertices*, and a set $E(G)$ of links between pairs of those points, that we call *edges* (see Figure 1.1). Seen another way, $E(G)$ is a set of pairs $e = \{u, v\}$ of vertices of G (we also write $e = uv$). In our case, your friends are the vertices, and there is an edge joining two

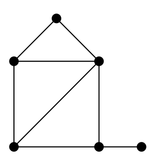

Figure 1.1: A graph with six vertices and eight edges.

of your friends if these two friends are on bad terms. We say that an edge e is *incident* to a vertex v if one of the endpoints of e is v, and we say that a vertex u is *adjacent* to a vertex v if there is an edge with endpoint u and v. Also, the vertices that are adjacent to u are the *neighbors* of u and the *degree* of u is the number of its neighbors.

In the setting of graphs, the problem we want to solve is called Vertex Cover. The goal is, given a graph G modeling the conflicts between your friends, to find a minimum number of vertices that can be deleted (along with their incident edges), such that the remaining graph has no edges. Such a set of vertices is called a *vertex cover*.

While the situation described here is just a toy example, graphs are used to simulate many real world problems. They can model the metro network in a big city, with vertices being the metro stations, and where two stations are adjacent if there is a direct connection between them. They can model molecules, where the vertices are the atoms, and the edges are the bonds between atoms.

They can again model social networks, with people represented by vertices, whose neighbors are their friends. Also, a phylogenetic tree is a specific kind of graph, called a *tree*.

Some well-known graphs

Let us already present well-known types of graphs (including trees). For $k \in \mathbb{N}$ ($\mathbb{N}$ is the set of non-negative integers), the *path* P_k is the graph with k vertices $v_1, \dots, v_k$ and edges $v_i vi + 1$ for $i \in [k-1] = \{1, \dots, k-1\}$ (see Figure 1.2). We say that P_k has length $k-1$ (i.e., the *length* of a

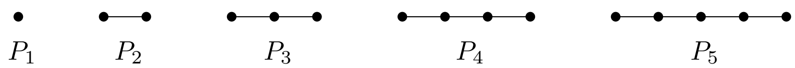

Figure 1.2: The paths of length zero to four.

path is its number of edges). For $k \geq 3$, the *cycle* C_k is the graph obtained from the path P_k by adding an edge $v_1 v_k$ (see Figure 1.3). For $t \in \mathbb{N}$, the *complete graph* K_t (alternatively the *clique* of

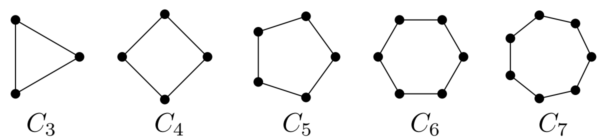

Figure 1.3: The cycles of size three to seven.

size t) is the graph with t vertices and all edges between vertices (see Figure 1.4). A graph is said to

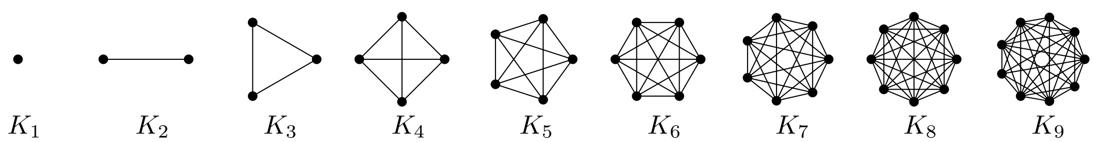

Figure 1.4: The complete graphs of size one to nine.

be *connected* if there is a path in G joining each pair of vertices of G. A *tree* is a connected graph with no cycle (see Figure 1.5). A *forest* is a graph (possibly disconnected) with no cycle. Thus, its connected components are trees. A *bipartite graph* is a graph whose edges can be partitioned into two sets A and B such that every edge has one endpoint in A and the other in B. In particular, paths, trees, forests, and cycles of even size are bipartite graphs. For $a, b \in \mathbb{N}$, the *complete bipartite graph* $K_{a,b}$ is the bipartite graph whose vertex set is the union of a vertex set A of size a and a vertex set B of size b and whose edge set is composed of all edges with one endpoint in A and the other in B (see Figure 1.5). A *planar graph* is a graph that can be drawn on a plane without edge

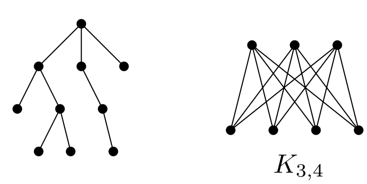

Figure 1.5: A tree and the complete bipartite graph $K_{3,4}$.

crossing. In particular, paths, cycles, trees, forests, and the graph of Figure 1.1 are planar graphs, but K_t is not for $t \geq 5$, and neither is $K_{a,b}$ for $a, b \geq 3$.

Algorithms

Let us come back to our birthday problem. Given that you (hopefully) still have many birthdays to come, and that your friends and their relationships will change over time, you want to create a set of instructions that you can run on your computer such that, given the graph of your friends' conflicts that you input, will output a minimum vertex cover of the graph. Such a set of instructions is what we call an *algorithm*.

Again, if you do not find our birthday setting very relatable, let us give another example of algorithm on graphs that you surely encountered in your everyday life. Assume you just arrive in a new city and you want to go from metro station a to metro station b, but you do not know the fastest way to do so. What do you do? Unless you enjoy reading the metro map to try to find the best way by yourself, you will open an application on your phone that, when you tell it you want to go from point a to point b, will output in a few seconds the fastest way to do so. As said above, the metro network can be modeled by a graph. What your application actually does is to solve Shortest Path on this graph. As its name indicates, the Shortest Path problem is the problem that takes as an input a graph and two vertices a and b and finds a path of shortest length between a and b.

1.1 Vertex Cover

Back to our birthday setting, the obvious way to create an algorithm solving Vertex Cover is to check, for each one of the 2^n subsets S of vertices of the graph G (n is the number of vertices of G), whether S is a vertex cover of G, and to output a smallest one. If, for now, you only have about ten friends (that is, $n = 10$), then the computer should be able to run the algorithm so that it outputs an answer quickly enough. However, your number of friends will grow over time, and when the time comes when you have more than 100 friends (which you expect will happen soon), the algorithm might terminate only after your birthday has passed. Here, the *running time* of your algorithm, i.e., the time the algorithm needs, in the worst case, to compute an answer, is $\mathcal{O}(2^n)$. We say that the algorithm runs in *exponential time*, given that the time needed to terminate increases exponentially with the number of vertices. This is why we would prefer to find an algorithm running in *polynomial time*, i.e., in time $\mathcal{O}(n^c)$ for some constant c, or, even better, in *linear time*, i.e., in time $\mathcal{O}(n)$.

NP-hardness

Unfortunately, Vertex Cover[1] is what we call an NP-hard problem [181], which means that, assuming the complexity assumption $\mathsf{P} \neq \mathsf{NP}$ (the question of P versus NP is one of the Millennium Prize problems), the problem cannot be solved in polynomial time. Researchers developed various methods to circumvent this issue. We can search for an approximate solution instead of the optimal one using approximation algorithms [311]. We can use heuristic algorithms [227], that work well in practice, but with no theoretical guarantee on the complexity. Using randomization, we can create non-deterministic algorithms [236] that solve the problem fast with high probability. Instead of computing the running time in the worst case, we can analyze the average running time [167]. Researchers even proposed more powerful computational models such as quantum or DNA computing [249, 315].

[1]To talk about NP-hardness, we need a *decision problem*, that is a problem that can be answered by yes or no. When considering Vertex Cover as a decision problem, it takes as an input a graph G and an integer $k \in \mathbb{N}$, and the question is whether G has a vertex cover of size at most k.

1.1.1 Parameterized complexity

Let us take a closer look at two such methods. The first one is *parameterized complexity* [74, 98, 119]. Parameterized complexity was introduced through the pioneering work of Downey and Fellows in the 90s [1, 93–96]. The key idea is that the hardness of a problem might originate from some more refined measure than just the size of the input (here the number n of vertices of the input graph). Theses measures are what we call *parameters*. If a parameter k is a the source of intractability for a problem, then we should be able to find an algorithm that is solvable in polynomial time after fixing k. Here, we consider *parameterized problems*, i.e., problems together with a parameter. For instance, VERTEX COVER, in its parameterized version, can be seen as the problem that, given a graph G and an integer $k \in \mathbb{N}$, outputs a minimum vertex cover of G of size at most k if it exists, and returns a no-answer otherwise.

XP-algorithms

In the birthday setting, you do not want to reject too many friends, since it will not be well-received. Maybe, if you need to reject more than k friends (for instance $k = 4$), then you just drop the idea and invite everyone, even though you know a fight will happen. In this case, an obvious algorithm is, for all subsets S of size at most k among your n friends, to check whether S is a vertex cover and to output a smallest such S. This algorithm runs in time $\mathcal{O}(n^{k+1})$. This is what we call an XP-*algorithm* (XP is "short" for slicewise polynomial), i.e., an algorithm running in time $f(k) \cdot n^{g(k)}$ for some functions f, g. For a fixed k, we indeed have a polynomial-time algorithm. However, because k is in the exponent of n, each fixed k gives rise to a different polynomial running time in n. In that sense, an XP-algorithm is a *non-uniform* algorithm with respect to k.

FPT-algorithms and branching

To remove this non-uniformity, we may search instead for a *fixed-parameter tractable algorithm* (FPT-*algorithm* for short), which is an algorithm running in time $f(k) \cdot n^c$, for some constant c and some function f. For each fixed k, the running time is $\mathcal{O}(n^c)$, which is hence uniform in k. Let us give such an example in our birthday setting, using a technique called *branching* [74]. Given that after deleting a vertex cover S, we delete all edges of the input graph G, this implies that, for each edge e of G, one of its endpoints is in S. Therefore, we can do the following procedure, with input a graph G and a $k \in \mathbb{N}$, where we search for a minimum vertex cover of G of size at most k. If $k < 0$, then there exists no vertex cover of size at most k. If G is edgeless, then the minimum vertex cover is the set $S = \emptyset$. Otherwise, there is at least one edge $e = \{u, v\}$ in G. A minimum vertex cover S contains at least one of u and v. Therefore, we run recursively the algorithm on $(G - u, k - 1)$ and $(G - v, k - 1)$ to obtain, if they exist, a minimum vertex cover S_u of $G - u$ of size at most $k - 1$ and a minimum vertex cover S_v of $G - v$ of size at most $k - 1$, respectively. If we did not find a vertex cover of size at most $k - 1$ for any of them, then there is no vertex cover of G of size at most k. If $|S_u| \leq |S_v|$, then $S_u \cup \{u\}$ is a minimum vertex cover of G, and otherwise, $S_v \cup \{v\}$ is a minimum vertex cover of G. Let us compute the running time of this algorithm. At each step of the recursion, we decrease k by one, and recursive calls happen only for $k \geq 0$, so the recursion has depth at most $k + 1$. Additionally, we make two recursive calls at each step of the recursion, so we do at most 2^{k+1} recursive calls in total. Given that a recursive call can be implemented in time $\mathcal{O}(n)$, the algorithm runs in time $\mathcal{O}(2^k \cdot n)$. We talk about *branching* because the procedure can be pictured as a search tree, whose root is the first edge $\{u, v\}$ we consider, with two children, one corresponding to the case where we pick u in the vertex cover, and the other to the case where we pick v in the vertex cover. We continue as such, picking a new edge at each node, and we stop when we arrive at depth

$k+1$, since we output a no-answer if a minimum vertex cover has size larger than k. We have two branches at each step and depth $k+1$, so the search tree has $\mathcal{O}(2^k)$ nodes.

W-hierarchy

Parameterized problems that admit FPT-algorithms (resp. XP-algorithms) form the parameterized complexity class FPT (resp. XP). Between FPT and XP, there is an entire hierarchy of classes expressing different levels of hardness of parameterized problems, called the W-*hierarchy* [97, 119], whose classes are ordered as follows:

$$\mathsf{FPT} = \mathsf{W}[0] \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{XP}.$$

If a parameterized problem is $W[t]$-hard for some $t \geq 1$, then it is unlikely that it can be solved in FPT-time. Many such problems exist. For instance, the k-Clique problem, which asks whether the input graph contains a complete graph on k vertices as a subgraph, is W[1]-complete when parameterized by k [96]. Similarly, k-Dominating Set, which asks for the existence of a set S of size at most k such that any vertex of the input graph is either in S or adjacent to a vertex in S, is W[2]-complete parameterized by k [95].

Additionally, if a parameterized problem is already NP-hard for some fixed value of the parameter, then it is said to be para-NP-hard. For instance, k-Coloring, which asks whether the vertices of the input graph can be colored using at most k colors, in such a way that no two adjacent vertices have the same color, is para-NP-hard parameterized by k, given that it is already NP-hard for $k = 3$ [181].

Kernelization

The running time to solve Vertex Cover can be optimized further using *kernelization*. Kernelization can be seen as a set of *preprocessing rules* that can be applied in polynomial time and that produce an equivalent instance whose size is bounded by a function of the parameter. For Vertex Cover, we can, for instance, observe that, for each vertex v of the input graph G, if v is not in a vertex cover, then all its neighbors are in this vertex cover. Therefore, if v has degree at least $k+1$, then any vertex cover of size at most k contains v. We can hence remove v and recurse on $(G-v, k-1)$, or, in other words, the instance (G, k) can be reduced to $(G-v, k-1)$. Additionally, if a vertex v has no neighbor, then it does not belong to any minimum vertex cover. Therefore, if a vertex v has degree zero, then the instance (G, k) can be reduced to $(G-v, k)$. With this two preprocessing rules, we can reduce in time $\mathcal{O}(k \cdot n)$ the instance (G, k) to an equivalent instance (G', k') such that $k' \leq k$ and such that every vertex in G' has degree at least one and at most k. Let S be a vertex cover of G' of size at most k'. Given that every vertex of G' has at least one neighbor, every vertex that is not in S is adjacent to a vertex in S. Therefore, given that each vertex in S has at most k neighbors and that S has size at most $k' \leq k$, there are at most k^2 vertices that are not in S. Therefore, if there is a vertex cover of G' of size at most k, then G' has at most $k + k^2$ vertices. Thus, if G' has strictly more that $k + k^2$ vertices, we can conclude to a no-instance, and otherwise, we obtained in polynomial time an instance (G', k') that is equivalent to (G, k), where $|V(G')| + k'$ is bounded by some function in k. This is what we call a *kernel*. Formally, a *kernelization algorithm* (*kernel* for short) for a parameterized problem Π is an algorithm that, given an instance (G, k), outputs, in polynomial time, an equivalent instance (G', k') of Π with $|V(G')| + k' \leq f(k)$, where f is a function. If f is more particularly a polynomial function, as is the case here, we talk about a *polynomial kernel*. Given a kernel for Π, we can obtain an FPT-algorithm for Π by solving the reduced instance exhaustively. Conversely, if there is an FPT-algorithm running in time $f(k) \cdot n^c$, then we can also obtain in time $\mathcal{O}(n^{c+1})$ a kernel of size $f(k)$ [74]. However, some problems that are in FPT are known to be unlikely to have a polynomial kernel [34].

To solve Vertex Cover given an instance (G, k), we can thus first apply our kernelization algorithm in time $\mathcal{O}(k \cdot n)$ to (G, k), which outputs an equivalent instance (G', k') with $|V(G')| = \mathcal{O}(k^2)$ and $k' \leq k$. Then, we apply our FPT-algorithm to (G', k'), which outputs in time $\mathcal{O}(2^{k'} \cdot |V(G')|) = \mathcal{O}(2^k \cdot k^2)$ a minimum vertex cover. We thus solve Vertex Cover in time $\mathcal{O}(2^k \cdot k^2 + k \cdot n)$.

From this trivial kernel, a long series of papers [21, 45, 58, 61, 80, 111, 220, 237, 238, 294] led to the current best kernel for Vertex Cover of [214], which has size $2k - c \log k$ for any constant c. Given that it has size $\mathcal{O}(k)$, we talk about a *linear kernel*. Concerning FPT-algorithms, a similarly impressive list of papers [21, 45, 54, 58, 242, 243, 296] led to the current best running time of $\mathcal{O}(1.2738^k + kn)$ [59]. Thus, if when you will have 100 friends instead of your current 10, the time your algorithm takes to terminate will only be multiplied by 10. If it seems like a lot to you, let us compare numbers: for $n = 10$ and $k = 4$, we have $2^n = 1024$ and $1.2738^k + kn \leq 43$, while for $n = 100$ and $k = 4$, we have $2^n \geq 10^{30}$ and $1.2738^k + kn \leq 403$. The multiplication by 10 concerning $1.2738^k + kn$ is nothing compared to the multiplication by 10^{27} concerning 2^n.

Of course, a downside is that we require k to be small for this to work. Still we may observe here that k does not need to be a constant to obtain a subexponential algorithm: for $k = \log n$, $\mathcal{O}(1.2738^k + kn) = \mathcal{O}(n^{\log(1.2738)} + n \log n)$ is polynomial. In particular in our birthday setting, rejecting at most $k = \log n$ friends seems reasonable.

1.1.2 Restricting the input

The second method we detail here to circumvent the issue of NP-hardness is to restrict the input. A problem might be NP-hard on complicated graph classes, but could become tractable on simpler classes. Instead of the class $\mathcal{G}_{\mathsf{all}}$ of all graphs, we may thus consider a more restrictive graph class, such as, for instance, the class of trees.

Trees

Back to our birthday setting, let us assume that the graph of conflicts of your friends forms a tree T. Let r be a vertex of T, that we call *root* of T. Then we say that (T, r) is a *rooted tree*, which is usually depicted (contrary to its name) with the root r at the top and the rest of the tree "pending" from r (see Figure 1.6 for an example). The vertices of a tree are sometimes called *nodes* of the

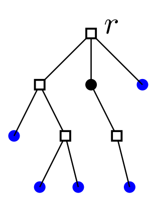

Figure 1.6: A tree T with root r. The leaves of (T, r) are depicted in blue, and the squared vertices form a vertex cover of T.

tree. The *leaves* are the vertices of degree one (aside from the root, which could also have degree one), and the other nodes are called *internal nodes*. The *parent* of a node $u \neq r$ is the node adjacent to u in the unique path between u and r, and the *children* of an internal node u are the neighbors of u aside from its parent. To find a minimum vertex cover of u, we can notice the following: if a vertex v has a unique neighbor u, then, for any vertex cover S containing v, $S \setminus \{v\} \cup \{u\}$ is also a vertex cover of size at most $|S|$. Therefore, given that the unique neighbor of a leaf v is its parent u, the set obtained by adding u to the minimum vertex cover of $T - u - v$ (the tree obtained after removing v and u from T) is a minimum vertex cover of T. As such, in a leaf-to-root manner, we may add the parents of leaves to the vertex cover, delete both the leaves and their parent from the tree, and recurse on the tree obtained after the deletion. This gives a minimum vertex cover such as the one depicted in Figure 1.6. We can thus solve Vertex Cover on trees in linear time.

Treewidth

Now, maybe the graph of your friends' conflicts is not a tree, but it might look similar to a tree when seen from afar. A way to measure how much a graph looks like a tree is a graph parameter called *treewidth*. A *graph parameter* is a function $\mathsf{p} : \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$, such as, for instance, the size $|V(G)|$ of a graph G. While it was (re)discovered several times since the 70s, the popularity of the parameter treewidth originates from its rediscovery in the Graph Minors series of Robertson and Seymour in the 80s [260], who demonstrated the usefulness of treewidth for problems related to minor-closed graph classes (see Section 1.3 and Section 3.1 for more on the subject). A *tree decomposition* of a graph G is a pair (T, β) composed of a tree T and a function β mapping each node t of T to a set $\beta(t)$ of vertices of G, called *bag* of t, such that each edge e and each vertex v belongs to some bag, and such that, for each vertex v, the nodes whose bag contain v form a subtree of T. Then, the *width* of (T, β) is the maximum size minus one of a bag, and the *treewidth* of G is the minimum width over all tree decompositions of G. See Figure 1.7 for an illustration. In particular,

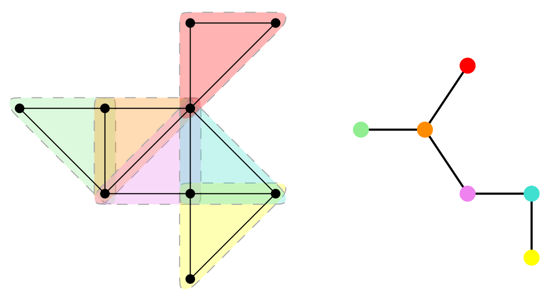

Figure 1.7: A graph of treewidth two (on the left) and an associated tree decomposition (T, β) of width two, where the tree T is represented on the right, and the bag $\beta(t)$ of each node t is the set of vertices of the same color as t on the left.

forests have treewidth one. There is a long line of research on the computation of the treewidth of a graph [13, 17, 26, 27, 33, 35, 110, 128, 131, 203, 204, 212, 255, 271]. In this thesis, we use most particularly the 2-approximation algorithm of Korhonen [203], running in time $2^{\mathcal{O}(\mathsf{tw})} \cdot n$. Treewidth is arguably the favorite parameter of researchers in parameterized complexity, mainly due to the fact that a large collection of problems on graphs is solvable in polynomial-time on graphs of bounded treewidth, or, said another way, is in FPT parameterized by tw [18, 67] (see Section 1.6) for more on the subject.

Dynamic programming

Let us see how to solve Vertex Cover on graphs of treewidth at most tw. Using for instance the 2-approximation algorithm of Korhonen [203], we find in time $2^{\mathcal{O}(\mathsf{tw})} \cdot n$ a tree decomposition of width at most $w = 2\mathsf{tw}$. This tree decomposition can then be transformed in a so-called *rooted nice tree decomposition* (cf. Section 4.3), which is a tree decomposition (T, β) with a root r and such that each node has at most two children. Then, we use a *dynamic programming technique* on (T, β, r). A dynamic programming technique consists in breaking the problem into a collection of simpler subproblems and in solving each subproblem only once by storing the solution for the next time the subproblem occurs. Our dynamic programming algorithm on (T, β, r) goes as follows. For each node t of T, in a leaf-to-root manner, and for each subset X of $\beta(t)$, we store the size $v_t(X)$ of the minimum vertex cover S_X containing X that occurs in the already processed graph G_t, i.e., the graph induced by the bag of t and its descendants (if there is no such vertex cover, we store ∞). Thus, a minimum vertex cover of G has size $\min_{X \subseteq \beta(r)} v_t(X)$. Notice that, by the connectivity of the bags containing a vertex in a tree decomposition, the vertices of G_t that are not in $\beta(t)$ will never appear again in the bag of a node between t and r. Let us see how to compute $v_t(X)$. For a leaf t, this is trivial given that $S_X = X$, if it is a vertex cover (otherwise, $v_t(X) = \infty$). For an internal node t with children t_1 and t_2, if X is a vertex cover of $\beta(t)$, then $v_t(X)$ is the minimum value $v_{t_1}(X_1) + v_{t_2}(X_2) - |X_1 \cap X| - |X_2 \cap X| + |X|$, over all X_1 and X_2 such that, for $i \in \{1, 2\}$, X_i is a subset of $\beta(t_i)$ such that $X \cap \beta(t_i) \subseteq X_i$. Given that a bag has size at most $w + 1$, there are at most 2^{w+1} subsets X for each node t. Thus, the computation of v_t can be done in time $2^{\mathcal{O}(\mathsf{tw})}$ and, given that we may assume that T has $\mathcal{O}(n)$ nodes (cf. Section 4.3), computing the size of a minimum vertex cover takes time $2^{\mathcal{O}(\mathsf{tw})} \cdot n$. It is then possible to apply some kind of backtracking algorithm going from the root to the leaves in order to deduce from the choices we made at each node such a minimum vertex cover.

1.2 Graph modification problems

Vertex Cover is the simplest of a category of problems called "graph modification problems". A *graph modification problem* is typically determined by

- a *target graph class* $\mathcal{H}$,
- a set of allowed *modifications* $\mathcal{M}$, and
- a *measure* p on the "modulator",

and the question is, given a graph G and an integer k, whether it is possible to transform G into a graph in $\mathcal{H}$ by applying modifications from $\mathcal{M}$, such that $\mathsf{p}(G, X) \leq k$, where $X \subseteq V(G)$ is the *modulator*, i.e., the set of all vertices that are modified or that are incident to a modified edge (here, the measure p depends on X, but might also depend on G). For Vertex Cover, the target class $\mathcal{H}$ is the class of edgeless graphs, the set of allowed modifications is $\mathcal{M} = \{\text{vertex deletion}\}$, and the measure is the size of the modulator X, i.e., $\mathsf{p}(G, X) = \mathsf{size}(X) = |X|$.

Starting from Vertex Cover, we may wonder what happens if we "grow" the target graph class, change the type of modification, or "grow" the size of the modulator by changing the measure on it. Each of these changes defines a new graph modification problem, and researchers study the parameterized complexity of those problems.

Beyond our toy example of birthday party to introduce Vertex Cover, graph modification problems have applications in domains as diverse as computational biology, computer vision, machine learning, networking, or sociology; see for instance [130, 292] and the references therein. One of the motivations behind the study of graph modification problems is linked to the concept of *distance from triviality* formalized by Guo, Hüffner, and Niedermeier [160], which expresses the closeness of a graph to a supposedly "simple" target graph class $\mathcal{H}$. Let us use Vertex Cover as an example. A problem Π is usually trivial on edgeless graphs. Therefore, we may hope that Π is also "easily solvable" when it is "close" to being edgeless, where the closeness here refers to the minimum size vc of a vertex cover, and "easily solvable" means solvable in FPT-time parameterized by vc. And this is actually the case for many problems such as the ones in the following papers [78, 112, 116, 117, 200, 215]. However, in order to apply those FPT-results parameterized by vc, we first need to be able to compute vc. The same holds for the other measures of distance from triviality. That is, assuming a problem Π is solvable in polynomial time on a graph class $\mathcal{H}$, Π might be solvable in FPT-time on graphs close to $\mathcal{H}$ parameterized by this closeness. Therefore, we need to be able to measure this closeness, which is done using graph modification problems.

Let us fix the measure to be the size of the modulator for now. Given a graph G and a set $S \subseteq V(G)$, $G - S$ is the graph obtained by removing the vertices of S from G. The most studied type of modification is arguably vertex deletion, for which one may define the following general graph modification problem, given some target graph class $\mathcal{H}$:

Vertex Deletion to $\mathcal{H}$

Input: A graph G, a $k \in \mathbb{N}$.
Question: Is there a set $S \subseteq V(G)$ of size at most k such that $G - S \in \mathcal{H}$?

Each different target class $\mathcal{H}$ defines a different graph modification problem. As presented above, when $\mathcal{H}$ is the class of edgeless graphs, then this is Vertex Cover. When $\mathcal{H}$ is the class of forests, then this is Feedback Vertex Set. When $\mathcal{H}$ is the class of bipartite graphs, then this is Odd Cycle Transversal.

One may also consider other modifications: if $\mathcal{M} = \{\text{edge addition}\}$ and $\mathcal{H}$ is the class of chordal graphs[2], then we obtain Chordal Completion; or even combine modifications: if $\mathcal{M} = \{\text{edge addition, edge deletion}\}$ and $\mathcal{H}$ is the class of graphs that are a union of cliques, then we obtain Cluster Edition. More generally, any pair composed of a fixed set of modifications $\mathcal{M}$ and a fixed target class $\mathcal{H}$ defines a new graph modification problem, hence their sheer number. For each such problem Π, the goal is hence to find an algorithm that solves Π as fast as possible.

Instead of studying each graph modification problem separately, an emerging trend is to find algorithmic *meta-theorems* solving many problems at once, given some restriction of the descriptive complexity of the problem via some logic and to the structure of the input via some graph parameter, with the objective to solve those problems efficiently (cf. Section 1.6 for more details). Of course finding a meta-theorem that would solve every problem efficiently is not realistic, as seen from the many hardness results [29, 30, 69, 122, 164, 216, 223, 291, 313]. Hence, two main lines of research can be distinguished regarding graph modification problems:

- **Efficiency:** solving a specific problem or certain families of problems as fast as possible.
- **Generality:** finding an algorithmic technique that can be applied to a general scheme for defining graph modification problems, often at the cost of its efficiency.

Following those lines of research, the objective in Part III of this thesis is to solve families of problems that can be solved by known algorithmic meta-theorems more efficiently. Let us discuss what "efficient" means here.

Graph modification problems that are known to be solvable in polynomial time do exist. One such example is Feedback Edge Set, where we ask for a minimum number fes of edges to delete from the input graph such that the resulting graph is a forest. fes is simply equal to $n + m + c$, where m is the number of edges, n the number of vertices, and c the number of connected components of the graph, which can be computed in polynomial time. Still, most graph modification problems are NP-hard [109, 216, 313].

[2] A *chordal graph* is a graph where every cycle of size at least four contains an edge connecting two non-consecutive vertices of the cycle.

Hence, we study in this thesis graph modification problems from the viewpoint of parameterized complexity. As presented in Section 1.1 for Vertex Cover, a rather natural parameter is the bound k on the measure on the modulator (usually the size), though we also sometimes use other graph parameters, such as the treewidth of the input.

The objective in Part IV of this thesis is to solve graph modification problems beyond the scope of known algorithmic meta-theorems. Before presenting those results (in Chapter 2), we first need to present a landscape of graph modification problems along with some known algorithmic results on those and the current limits.

As said above, a graph modification problem is defined from three ingredients: a target class $\mathcal{H}$, a set of allowed modifications $\mathcal{M}$, and a measure p on the modulator. We can also see it as a *modulator versus target* scheme: on the one hand, we have whatever is modified (the modulator) characterized by $\mathcal{M}$ and p, and on the other hand, we have whatever remains (the target) characterized by $\mathcal{H}$. We thus proceed as follows for the next three sections. In Section 1.3, we consider other target graph classes, beyond edgeless graphs, and we more generally explain how to encompass several target classes at once by excluding forbidden patterns. In Section 1.4, we focus on modifications and introduce replacement actions, a way to generalize several types of modifications. In Section 1.5, we discuss measures that have been considered, other than the size of the modulator, in the literature.

1.3 Target class

As said above, the problem of Vertex Cover corresponds to Vertex Deletion to $\mathcal{H}$ when $\mathcal{H}$ is the class of edgeless graphs. Let us see what happens when we change the target graph class.

1.3.1 Vertex Deletion to $\mathcal{H}$

If $\mathcal{H}$ is the class of forests, then we obtain Feedback Vertex Set. In Subsection 1.1.2, we mention that treewidth is a parameter measuring how much a graph looks like a tree or, more generally, a forest. The minimum size fvs of a *feedback vertex set* of a graph G (a set S such that $G - S$ is a forest) is another graph parameter measuring how close G is to being a forest. Given that many problems are solvable in polynomial time on forests, many problems are also solvable in FPT-time parameterized by fvs - see for instance [38, 172, 207, 226, 234]. Just as is the case for Vertex Cover, Feedback Vertex Set has been extensively studied parameterized by the size k of the modulator. An impressive list of papers [25, 32, 50, 57, 76, 79, 94, 159, 180, 201, 252, 253, 306] leads to the current best running time of $\mathcal{O}(2.7^k \cdot n)$ [217] (the linearity comes from [170]). On the kernalization side, the best kernel for Feedback Vertex Set has size $2k^2 + k$ [170].

When $\mathcal{H}$ is the class of bipartite graphs, we obtain Odd Cycle Transversal, whose parameterized complexity by the size k of the modulator has been studied for instance in [175, 192, 225, 254].

When $\mathcal{H}$ is the class of planar graphs, the problem is often called Planarization and the currently fastest algorithm runs in time $2^{\mathcal{O}(k \log k)} \cdot n$ [176].

Surfaces

Planar graphs are the graphs that can be *embedded* in the sphere (or equivalently the plane), i.e., graphs that can be drawn in the sphere with no edge crossings. More generally, we may consider the class $\mathcal{G}_\Sigma$ of graphs that can be embedded in some surface Σ. Let us give some intuition on surfaces (we refer the interested reader to [233] for more on the subject).

We call here *surface* a compact and connected *2-manifold*, which is a topological space such that every point has a neighborhood that is topologically equivalent to an open disk and any two distinct points can be contained by disjoint neighborhoods. The simplest surface is the *sphere*, and the *surface classification theorem*, that was first proved rigorously in the 1920s by Brahana [42], essentially states that every surface can be obtained from the sphere by adding h "handles" and c "crosscaps", for some $c, h \in \mathbb{N}$ (see Figure 1.8).

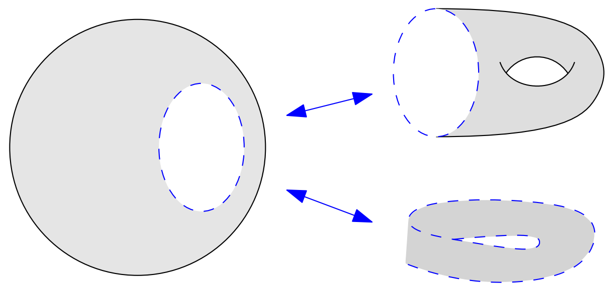

Figure 1.8: Adding a handle or a crosscap to the sphere: the dashed boundaries are glued together.

Adding a *handle*, essentially means adding a hole to the surface. The *torus* (the surface with one hole that looks like a donut) is obtained from the sphere by adding one handle, and more generally, the *k-torus* (the surface with k holes) is obtained from the sphere by adding k handles (cf. Figure 1.9). Those surfaces, that can be obtained from the sphere by adding only handles, are called *orientable surfaces*. They are easy to picture given that they can be represented in 3D. The number of handles of an orientable surface is called its *genus*.

The other surfaces, that require to add "crosscaps", are called *non-orientable surfaces*. A *crosscap* is essentially a *Möbius strip*, i.e., the half-twisted strip pictured in Figure 1.8. Notice that the boundary of a Möbius strip is a unique cycle. Adding a crosscap to a surface Σ essentially consists in cutting Σ along a cycle C bounding a disk Δ, and replacing Δ with a Möbius strip whose boundary is glued to C. By adding one crosscap to the sphere, we obtain the *projective plane*. The projective plane, along with the other non-orientable surfaces, is not representable in 3D, though it can be represented in 4D. Still an attempt to picturing it is proposed in Figure 1.9. By adding another crosscap, we obtain the Klein bottle. This is arguably the most famous non-orientable surface, perhaps because it is relatively easy to depict. With yet another crosscap, we obtain Dyck's surface. The other non-orientable surfaces are usually designated by their number of handles and crosscaps.

An important result on surfaces is Dyck's theorem [103, 132, 302]. It states that two crosscaps are equivalent to a handle under the presence of a third crosscap. This implies that Dyck's surface is equivalently the surface obtained from the sphere by adding one handle and one crosscap. More generally, it implies that any surface can be obtained from the sphere by adding h handles and c crosscaps, where $h \in \mathbb{N}$ and $c \in \{0, 1, 2\}$.

Instead of the genus of a surface Σ, which mainly refers to orientable surfaces, we prefer to use a variant that is in accordance with the fact that two crosscaps are essentially equal to a handle. This variant is the *Euler genus* of Σ, which is defined as $2h + c$, where Σ has h handles and c crosscaps. Hence, for Dyck's surface for instance, no matter whether we say that it has three crosscaps or one handle and one crosscap, the Euler genus is three. Interestingly, if g is the smallest integer such that a graph G embeds in a surface Σ of Euler genus g, then $v + f - e = 2 - g$, where $v = |V(G)|$, $e = |E(G)|$, and f is the number of connected components obtained after cutting Σ along the edges of G (the number of *faces*). The value $2 - g$ is called *Euler characteristic* of a surface (in terms of *genus* g of an orientable surface, the Euler characteristic is $2 - 2g$).

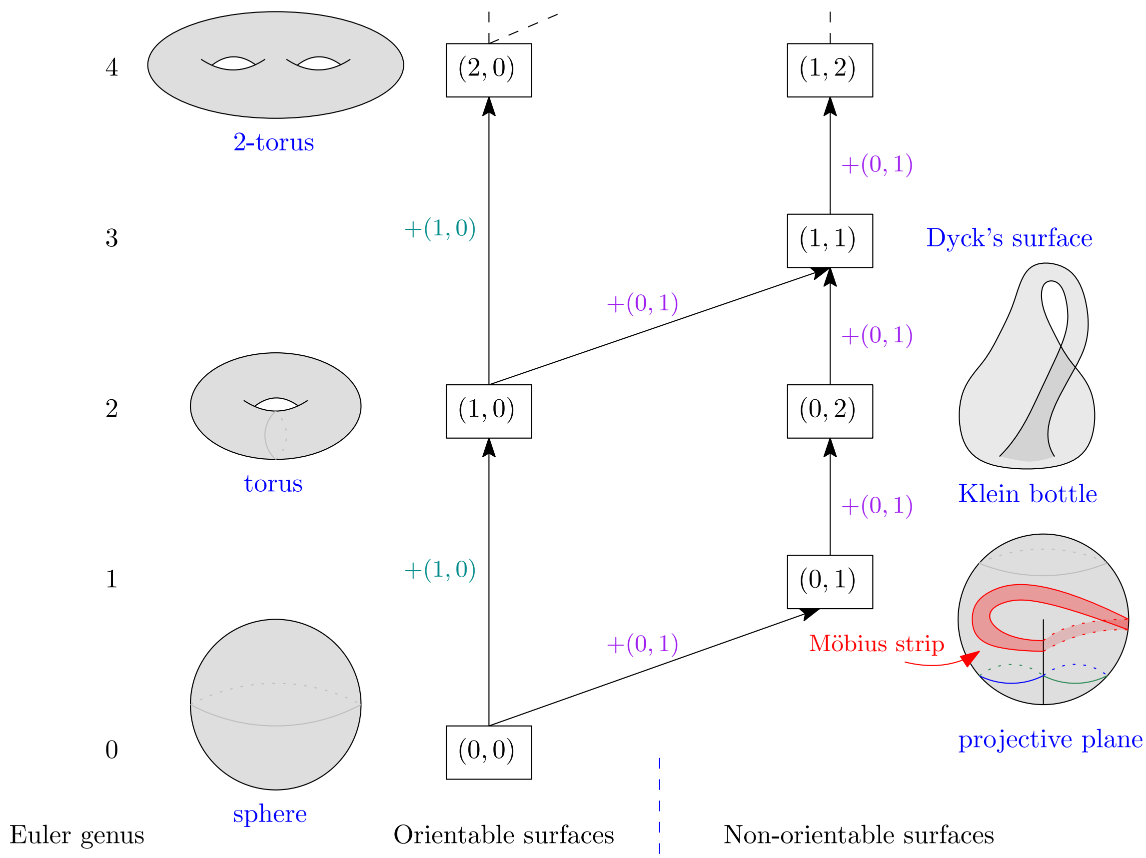

Figure 1.9: Representation of the surfaces of small Euler genus. Each surface is characterized by a pair (h, c) where $h \in \mathbb{N}$ is the number of handles of the surface and $c \in \{0, 1, 2\}$ is its number of crosscaps.

Back to our graph modification problems, given that every surface locally looks like a disk, results on planar graphs are often generalizable to surfaces (parameterized by the Euler genus). This is the case for Vertex Deletion to $\mathcal{H}$: when $\mathcal{H} = \mathcal{G}_\Sigma$ for some surface Σ of Euler genus g, then there is an algorithm running in time $2^{\mathcal{O}_g(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$[3] [202].

1.3.2 Excluding forbidden patterns

In Subsection 1.3.1, we mainly present results on particular target graph classes. Due to the sheer number of graph classes, enumerating results on each of them one by one would be a daunting task. Instead, we would prefer to find a property $\mathcal{P}$ on graph classes such that graph modification problems to target classes with this property are "easily" solvable. One such method is to characterize target graph classes via the exclusion of forbidden graphs according to some partial order on graphs. Let us give some notations concerning partial orders on graphs.

[3]Given some $t \in \mathbb{N}$ and some functions $f, g : \mathbb{N} \to \mathbb{N}$, we write $g(n) = \mathcal{O}_t(f(n))$ in order to denote that there exists some function $h : \mathbb{N} \to \mathbb{N}$ such that $g(n) \leq \mathcal{O}(h(t) \cdot f(n))$.

Partial orders. Let $\preceq$ be a partial order on graphs[4]. We say that a graph class $\mathcal{H}$ is *closed under* $\preceq$ if, for each $G \in \mathcal{H}$ and for each $H \preceq G$, we have $H \in \mathcal{H}$. Then, we can define the set of $\preceq$*-obstructions* of $\mathcal{H}$ to be the (possibly infinite) set $\mathsf{obs}_{\preceq}(\mathcal{H})$ of graphs F such that $F \notin \mathcal{H}$ and, for each $G \preceq F$, $G \in \mathcal{H}$. A standard way to describe many target classes at once is to ask for $\mathcal{H}$ to be any target class such that $\mathsf{obs}_{\preceq}(\mathcal{H})$ has some particular properties, such as being finite or containing a planar graph, or more generally, for $\mathcal{H}$ to be closed under $\preceq$.

Let us mention some of the most well-known partial orders on graphs (the operations mentioned below are illustrated in Figure 1.10).

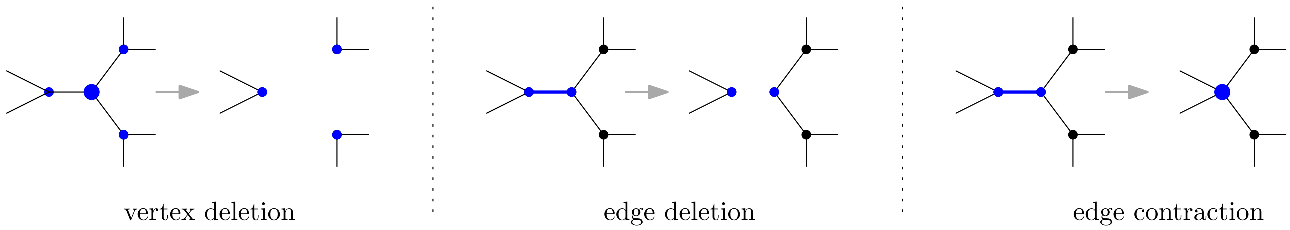

Figure 1.10: Illustration of a vertex deletion, an edge deletion and an edge contraction.

(Induced) subgraphs

One of the most natural partial orders on graphs is through the deletion of vertices and/or edges. A graph H is an *induced subgraph* of a graph G, denoted by $H \preceq_{\mathsf{i}} G$, if H can be obtained from G by removing vertices. A graph class that is closed under induced subgraphs is said to be *hereditary*. A graph H is a *subgraph* of a graph G if H can be obtained from G by removing edges and vertices. A graph class that is closed under subgraphs is said to be *monotone*. Graph modification problems where the target class is hereditary or monotone have been extensively studied from the parameterized complexity viewpoint. Most notably, if $\mathcal{H}$ is graph class such that $\mathsf{obs}_{\preceq_{\mathsf{i}}}(\mathcal{H})$ is finite, then Cai [47] proved that, given a graph G and three integers i, j, k, one can decide in FPT-time (parameterized by i, j, and k) whether one can remove at most i vertices, remove at most j edges, and add at most k edges so that the modified graph belong to $\mathcal{H}$. See also for instance [7, 16, 40, 48, 75, 174, 228, 239, 283] and, more particularly, the extensive survey of [69] (restricted to modifications related to edges).

Subdivisions/topological minors

Another rather natural partial order on graphs is to say that a graph H is smaller than a graph G if H can be "drawn" in G, i.e., if each vertex v of H is mapped to a vertex x_v of G and each edge $\{u, v\}$ of H is mapped to a path P_{uv} in G with endpoints x_u and x_v, where the vertices x_v are pairwise distinct and the paths P_{uv} are pairwise internally vertex-disjoint. We say that G contains a *subdivision* of H as a subgraph (a subdivision of a graph H, which corresponds to replacing edges by paths, is depicted in Figure 1.11). Equivalently, a graph H is a *topological minor* of a graph G, denoted by $H \preceq_{\mathsf{tm}} G$, if H can be obtained from G by removing edges, removing vertices, and contracting edges that have at least one endpoint of degree two. Modification problems to topological-minor-closed graph classes have mainly been investigated for vertex deletion problems, see for instance [6, 22, 126, 145].

[4] A *partial order* $\preceq$ on a set S is a binary relation that is reflexive ($a \preceq a$ for each $a \in S$), antisymmetric ($a \preceq b$ and $b \preceq a$ implies $a = b$), and transitive ($a \preceq b$ and $b \preceq c$ implies $a \preceq c$).

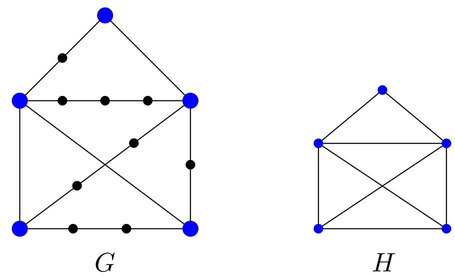

Figure 1.11: The graph G is a subdivision of the graph H.

Minors

Let us now present a partial order with nice properties.

A graph H is a *minor* of a graph G, denoted by $H \preceq_{\mathsf{m}} G$, if H can be obtained from G by removing edges, removing vertices, and contracting edges (see Figure 1.12). A graph class that is closed under minors is said to be *minor-closed*. As shown in Figure 1.16, minor-closed graph classes

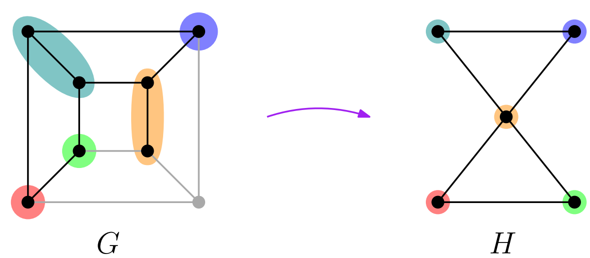

Figure 1.12: A minor H of a graph G.

are also topological-minor-closed, monotone, and hereditary. Among the classes mentioned above, the classes of edgeless graphs, of forests, of planar graphs, and of graphs embeddable in a surface Σ are minor-closed. Bipartite graphs however, form a monotone (and thus hereditary) graph class, but that is not (topological-)minor-closed.

Let us discuss some nice properties of minor-closed graph classes.

Well-quasi-order. Kuratowski proved in the 30s that the topological-minor-obstructions of the class of planar graphs are K_5 and $K_{3,3}$ [211]. Later, Wagner proved another variant of this result: the minor-obstructions of the class of planar graphs are K_5 and $K_{3,3}$ [309] (see Figure 1.13). Given that

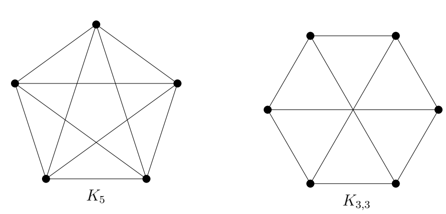

Figure 1.13: The (topological-)minor-obstructions of planar graphs.

planar graphs form a minor-closed graph class, this gave rise to the so-called Wagner's conjecture, which states that every minor-closed graph class has a finite number of minor-obstructions. In more technical terms, it states that minor containment is a *well-quasi-order* on graphs. A *well-quasi-order*

(WQO) on graphs is a partial order such that there is no infinite set of graphs that are pairwise incomparable by the partial order. The other partial orders defined above, that is (induced) subgraph containment and topological minor containment, are not WQOs given that there exist some infinite set $(S_k)_{k\in\mathbb{N}}$ of pairwise incomparable graphs for those partial orders (cf. Figure 1.14).

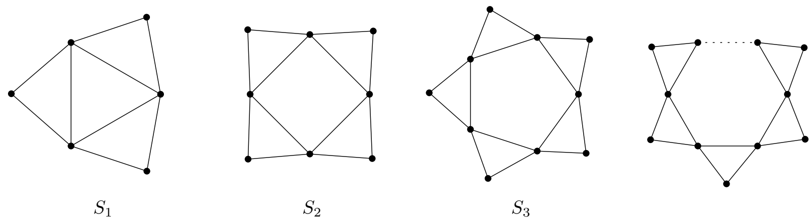

Figure 1.14: The family $(S_k)_{k\in\mathbb{N}}$ is a family of pairwise incomparable graphs for (induced) subgraph and topological minor containment.

The Graph Minors series. In the span of 23 papers [258–267, 269–281], Robertson and Seymour proved several results on minors and developed many techniques that are at the base of the Graph Minor Theory (see Chapter 3 for more on those techniques). Their primary objective was to prove Wagner's conjecture, which they achieve in [278]. Additionally, the main algorithmic contribution of the Graph Minors series is an algorithm that solves in time $\mathcal{O}_h(n^3)$[5] the Minor Containment problem which asks whether a fixed graph H on h vertices is a minor of an input graph G on n vertices [271]. The cubic running time has since then been improved to $\mathcal{O}_h(n^2)$ by Kawarabayashi, Kobayashi, and Reed [188], and more recently to $\mathcal{O}_h(n^{1+o(1)})$ by Korhonen, Pilipczuk, and Stamoulis [205]. An algorithmic consequence of the above concerns the Membership in $\mathcal{H}$ problem that asks whether the input graph G belongs to a fixed graph class $\mathcal{H}$: if $\mathcal{H}$ is minor-closed, then Membership in $\mathcal{H}$ is equivalent to checking whether G excludes as minors all the minor-obstructions of $\mathcal{H}$. Due to the finiteness of the minor-obstructions of $\mathcal{H}$, Membership in $\mathcal{H}$ thus reduces to applying a finite number of times an algorithm solving Minor Containment. Therefore, it is solvable in time $\mathcal{O}_{s_{\mathcal{H}}}(n^{1+o(1)})$ where $s_{\mathcal{H}}$ is the size of the largest obstruction of $\mathcal{H}$.

A meta-theorem. When the yes-instances (or the no-instances) of a decision problem Π form a minor-closed graph class $\mathcal{G}$, it implies that solving Π reduces to solving Membership in $\mathcal{G}$, which can be done in polynomial time. Under this viewpoint, the Graph Minors series build a framework to solve a wide family of problems in polynomial time. This is hence a so-called *meta-algorithmic theorem*.

In particular, this applies to graph modification problems. For instance, when the target class $\mathcal{H}$ is minor-closed, then the set $\mathcal{A}_k(\mathcal{H})$ of all graphs G such that (G, k) is a yes-instance of Vertex Deletion to $\mathcal{H}$ is also a minor-closed graph class. Therefore, this immediately implies that Vertex Deletion to $\mathcal{H}$ is solvable in time $\mathcal{O}_{s_{\mathcal{H}},k}(n^{1+o(1)})$.

Interestingly, Robertson and Seymour's proof of Wagner's conjecture is not constructive: while they prove that the obstruction set of minor-closed graph classes is finite, they do not provide a way to construct it. Thus, they only prove the *existence* of a polynomial algorithm for Membership in $\mathcal{H}$ when $\mathcal{H}$ is minor-closed. Actually, Friedman, Robertson, and Seymour proved that there exists no proof yielding a way to construct the minor-obstruction set $\mathsf{obs}(\mathcal{H})$ for each minor-closed graph

[5]Given some $t \in \mathbb{N}$ and some functions $f, g : \mathbb{N} \to \mathbb{N}$, we write $g(n) = \mathcal{O}_t(f(n))$ in order to denote that there exists some function $h : \mathbb{N} \to \mathbb{N}$ such that $g(n) \leq \mathcal{O}(h(t) \cdot f(n))$.

class $\mathcal{H}$ [133], while Fellows and Langston proved that there is no algorithm that, given a finite description of a minor-closed graph class $\mathcal{H}$, outputs $\mathsf{obs}(\mathcal{H})$ [113].

Constructing obstructions. Hence, a line of research in Graph Minor Theory consists in constructing the obstructions of minor-closed graph classes, which immediately implies an algorithm solving membership to the said graph class. In particular, given a "simple" minor-closed graph class $\mathcal{H}$ whose obstructions are supposed to be known, researchers study mechanisms for constructing obstructions of minor-closed graph classes that are "close" to $\mathcal{H}$ in the sense of the distance from triviality of Guo, Hüffner, and Niedermeier [160]. For instance, given the set $\mathsf{obs}(\mathcal{H})$ of obstructions of a minor-closed graph class $\mathcal{H}$, the constructibility of the obstructions of the class $\mathcal{A}_k(\mathcal{H})$ obtained by adding at most k vertices to graphs of $\mathcal{H}$ as been extensively studied [3, 91, 92, 115, 125, 230, 244, 316]. Other modification operations have also been investigated for instance in [43, 55, 89, 114, 161, 213, 290].

One of the goals of this thesis (although not necessarily the main one) is thus the following.

Bound the size of the obstructions of minor-closed graph classes composed of the yes-instances of some graph modification problems.

We do so in particular in Chapter 6 and Chapter 8. Those results are presented in Section 2.2 and Section 2.4, respectively.

Vertex Deletion to Minor-closedness. The Graph Minor series of Robertson and Seymour gave rise to many researches on problems related to minors and, in particular, explains why graph modification problems where the target class is minor-closed are extensively studied. Studies on modification problems to (specific) minor-closed graph classes include for instance the following [3, 57, 59, 124, 125, 176, 184, 197, 202, 229]. Concerning Vertex Deletion to $\mathcal{H}$, when $\mathcal{H}$ is any minor-closed graph class, the best parametric dependence is obtained in [284] with a running time of $2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}} \cdot n^3$. This parametric dependence can be improved if we add some constraints on $\mathsf{obs}(\mathcal{H})$. If an obstruction of $\mathcal{H}$ is planar, then Vertex Deletion to $\mathcal{H}$ can be solved in time $2^{\mathcal{O}(k)} \cdot n^2$ [197]. If we additionally ask that all obstructions are connected, then the running time drops to $2^{\mathcal{O}(k)} \cdot n$ [125]. Asking for an obstruction to be planar is a rather restrictive condition. To relax this condition, we can instead ask that an obstruction F of $\mathcal{H}$ is close to being planar. By this, we mean that F is an *apex graph*, i.e., F contains a vertex v such that $F - v$ is planar. In this case, we say that $\mathcal{H}$ is an *apex-minor-free* graph class. Both K_5 and $K_{3,3}$ are apex graphs, so planar graphs are apex-minor-free. More generally, the class $\mathcal{G}_\Sigma$ of graphs embeddable in a surface Σ is apex-minor-free. Indeed, there is $k_\Sigma \in \mathbb{N}$ such that $K_{3,k_\Sigma} \notin \mathcal{G}_\Sigma$ [233, Theorem 4.4.7], which is an apex graph, given that K_{2,k_Σ} is a planar graph. When $\mathcal{H}$ is any apex-minor-free graph class, the authors of [284] obtain a running time of $2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}} \cdot n^2$.

Odd-minors

Let us mention a last partial order on graphs that could imply results as promising as the ones on minor-closed graph classes.

A graph H is an *odd-minor* of a graph G if H can be obtained from G by removing edges, removing vertices, and contracting edge cuts[6] (see Figure 1.15 for an illustration). Another way to see it is that an odd-minor is a minor that preserves the parity of the length of cycles (cf. Section 9.2 for

[6] An edge set E' is an *edge cut* in a graph G if there exists a partition of $V(G)$ into two sets A and B such that E' is exactly the set of edges of G with one endpoint in A and the other in B.

more on the subject). A graph class that is closed under odd-minors is said to be *odd-minor-closed*. In

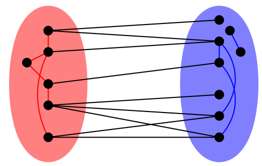

Figure 1.15: The black edges form an edge cut.

particular, the class of bipartite graphs is odd-minor-closed, with unique odd-minor-obstruction K_3. Hence, a minor-closed graph class is also odd-minor-closed, but the contrary does not hold, given that the class of bipartite graphs is not minor-closed. Other examples of odd-minor-closed graph classes are the classes of graphs excluding odd (resp. even) cycles of big size. For instance, the K_{2p+1}-odd-minor-free graphs are exactly the graphs with no odd cycles of size $2p+1$ or more.

Hadwiger's conjecture [163], which is open since 1943, states that if a graph excludes K_t as a minor, then its chromatic number[7] is at most $t-1$. In 1993, Gerards and Seymour [177] generalized this conjecture to odd-minors, hence drawing attention to odd-minors: the Odd Hadwiger's conjecture states that if a graph excludes K_t as an odd-minor, then its chromatic number is at most $t-1$. Note that a similar conjecture when K_t is excluded as a topological minor was disproved by Catlin in [51]. Since then, several papers regarding odd-minors appeared. Most of them focus on the resolution of the Odd Hadwiger's conjecture (see for instance [137], and [297] for a nice overview of the results), while some others aim at extending the results of Graph Minor Theory to odd-minors (see for instance [85, 168, 193]).

Similarly to minors, Odd-Minor Containment, that asks whether a fixed graph H is an odd-minor of the input graph G, is solvable in FPT-time parameterized by the size of H [193] (unfortunately, only a short version of this paper exists to our knowledge, see also [168] for an XP-result on the subject). Moreover, as explained by Huynh in [168], Geelen, Gerards, and Whittle claimed in 2009 that odd-minor containment is a WQO, though no proof as been written since then to our knowledge. If these results are true, then they would imply, as is the case for minors, that Membership in $\mathcal{H}$ is solvable in FPT-time for any odd-minor-closed graph class $\mathcal{H}$. In particular, given that $\mathcal{A}_k(\mathcal{H})$ is odd-minor-closed for any odd-minor-closed graph class $\mathcal{H}$, this would imply the existence of an FPT-algorithm solving Vertex Deletion to $\mathcal{H}$ for any odd-minor-closed graph class $\mathcal{H}$.

The Graph Odd-minor Theory is unfortunately far less developed than the Graph Minor Theory. Therefore, not much is known about modification problems to any odd-minor-closed graph class (that is not minor-closed), other than the vertex deletion to bipartite graphs, that is, Odd Cycle Transversal [192, 254].

A hierarchy of the graph classes closed by the partial orders defined above is depicted in Figure 1.16. Other partial orders exist, such as *immersions* [139, 140], though we will not go into details about them here.

In this section, we mainly talked about vertex deletion problems. This is because vertex deletion is one of the simplest modification operations, and thus, one of the most extensively studied. While we develop more on the other modification operations in the next section, let us already present one of the objectives of this thesis.

[7]The *chromatic number* of a graph G is the minimum number of colors necessary to color the vertices of G such that no two adjacent vertices have the same color.

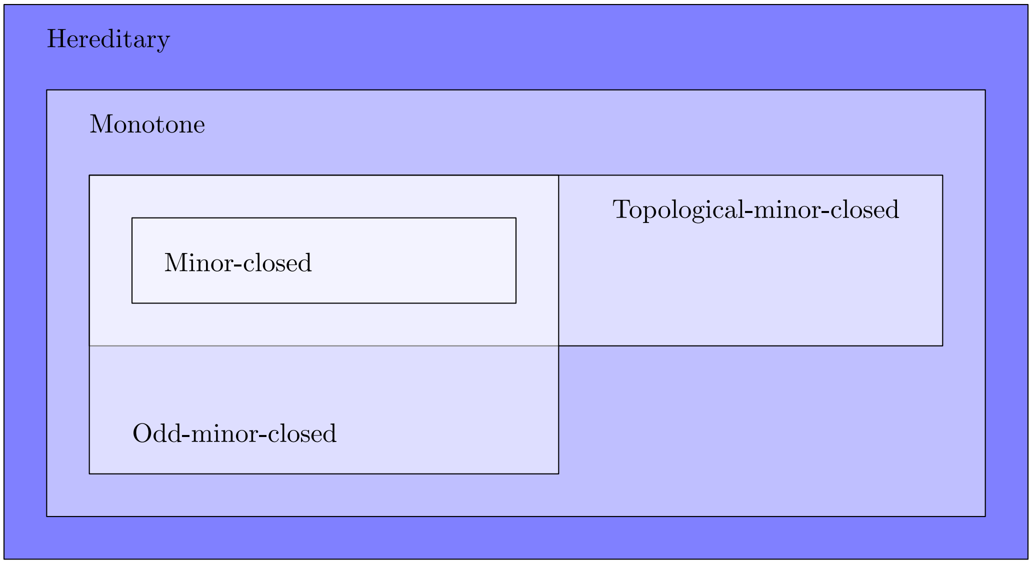

Figure 1.16: Hierarchy of graph classes properties: a minor-closed graph class is both odd-minor-closed and topological-minor-closed, and all of those are monotone graph classes and, more generally, hereditary graph classes.

> Extend the results on graph modification problems where the modification is vertex deletion to other modification operations.

We do so in particular in Chapter 5, Chapter 6, and Chapter 7. See Section 2.1, Section 2.2, and Section 2.3 for a description of those results.

1.4 Modification

For Vertex Cover, the modification is *vertex deletion*. While this is certainly the most studied modification, as discussed in the previous section, other modification operation do exist. Some of the most studied are *edge deletion* and *edge addition*. If we allow both, then we get *edge edition*. We refer the reader to the extensive survey of [69] concerning the parameterized complexity of Edge Deletion/Addition/Edition to $\mathcal{H}$. Let us also mention the *edge contraction* operation (see [147] for an overview on Contraction to $\mathcal{H}$). Some more exotic modification operations exist, such as the deletion of a connected set, the deletion of an independent set, the complementation of a subgraph, the deletion or contraction of a matching, and so on [39, 66, 73, 121, 218, 221, 251].

In order not to study each modification operation one by one, it is necessary to define a general framework that can describe many of them. One of them uses replacement actions, introduced by Fomin, Golovach, and Thilikos in [121].

Replacement actions

The idea of replacement actions is not to consider each modification operation, such as deleting an edge, separately, but instead to consider the modulator in its whole, for instance, the minimum induced subgraph of the input graph where k edges will be removed. More formally, Fomin, Golovach, and Thilikos defined in [121] a *replacement action* as a function $\mathcal{L}$ that maps each graph H to a

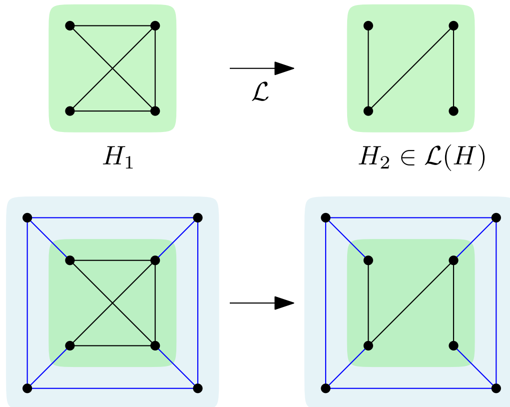

Figure 1.17: Illustration of a graph modification allowed by the replacement action $\mathcal{L}$.

collection of graphs of size $|V(H)|$ (cf. Figure 1.17 for an illustration) corresponding to the "allowed modifications". Given a target class $\mathcal{H}$, $\mathcal{L}$-Replacement to $\mathcal{H}$ is the problem where the input is a graph G and the question is whether it is possible to replace some induced subgraph H_1 of G on at most k vertices by a graph H_2 in $\mathcal{L}(H_1)$ so that the resulting graph belongs to $\mathcal{H}$. Defined as such, $\mathcal{L}$-Replacement to $\mathcal{H}$ can simulate any problem that is a combination of edge removal and edge addition of bounded size, such as whether it is possible to remove a matching of size at most k to belong to $\mathcal{H}$, or whether one can find a set X of size at most k and replace the subgraph induced by X by its complement, so that the modified graph is in $\mathcal{H}$. Fomin, Golovach, and Thilikos proved in [121] that $\mathcal{L}$-Replacement to $\mathcal{H}$ is solvable in time $f(k) \cdot n^2$ for some computable function f when $\mathcal{H}$ is the class of planar graphs.

A limit of this framework of replacement actions is that it encompasses only modifications related to edge additions and deletions. It does not encompass the deletion of vertices, or the contraction of edges, for instance. Additionally, $\mathcal{L}$-Replacement to $\mathcal{H}$ has only been studied when $\mathcal{H}$ is the class of planar graphs and no explicit parametric dependency is known. We hence have the following objective.

> Simultaneously extend the framework of replacement actions so that it encompasses more modification operations and solve $\mathcal{L}$-Replacement to $\mathcal{H}$ on more general graph classes and with moderate parametric dependencies.

This is what we do in Chapter 7 (see Section 2.3 for a description of the results).

1.5 Measure on the modulator

For Vertex Cover, and, more generally, for most graph modification problems mentioned above, the measure on the modulator X is $\mathsf{size}(X) = |X|$. However, nothing stops us from quantifying the modulator using another more refined parameter. For instance, we could ask whether the input graph G has a modulator X such that $G[X]$[8] has treewidth at most w and $G - X$ is planar. Given a problem Π that is solvable in polynomial time on graphs of bounded treewidth, and on planar

[8]Given a graph G and $X \subseteq V(G)$, the graph *induced* by X, denoted by $G[X]$, is the induced subgraph of G with vertex set X.

graphs, we could hope that Π is solvable in FPT-time parameterized by w on such graphs. Given that the class $\mathcal{H}_w$ of such graphs is more general that the class $\mathcal{A}_w(\mathcal{P})$ of graphs that are w vertices away from being planar, solving Π on $\mathcal{H}_w$ is more general than solving Π on $\mathcal{A}_w(\mathcal{P})$. Unfortunately, Farrugia [109] proved that, if $\mathcal{G}$ and $\mathcal{H}$ are two hereditary graph classes that are closed under disjoint union[9], then checking whether there is $X \subseteq V(G)$ such that $G[X] \in \mathcal{G}$ and $G - X \in \mathcal{H}$ is NP-complete unless both $\mathcal{G}$ and $\mathcal{H}$ are the class of edgeless graphs. However, there are some positive results if, instead of a measure on the graph $G[X]$ induced by X, we consider the "torso" of X in G.

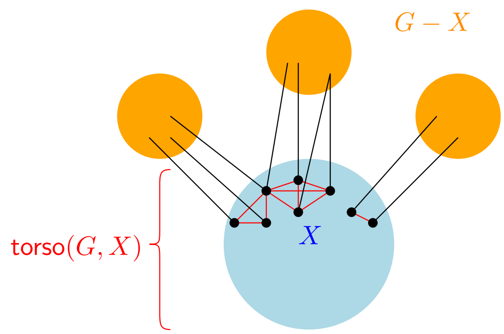

Figure 1.18: Torso of a set X in a graph G. For each connected component C of $G - X$ (in orange), we add all edges (in red) between pairs of vertices of X adjacent to vertices in C.

Torso. The *torso* of a set $X \subseteq V(G)$ in a graph G, denoted by $\mathsf{torso}(G, X)$, is the graph obtained from $G[X]$ by making a clique out of the neighborhood of each connected component of $G - X$ (see Figure 1.18). Compared to $G[X]$, $\mathsf{torso}(G, X)$ keeps information about each connected component C of $G - X$ that has been removed and, more specifically, about the paths going through C with endpoints in X.

Let us present a first graph modification problem whose measure on the modulator X is not its size, but a different parameter applied to the torso of X. For this, we need to define the graph parameter treedepth.

Treedepth. Similarly to treewidth, the *treedepth* of a graph G, denoted by $\mathsf{td}(G)$, is a minor-monotone graph parameter with $\mathsf{tw} \preceq \mathsf{td} \preceq \mathsf{size}$ that has been (re)discovered several times since the 70s, though its popularity as well as the term are due to Nešetřil and Ossona de Mendez [240, 241]. The most common definition of treedepth might be the following:

$$\mathsf{td}(G) = \begin{cases} 0 & \text{if } G \text{ is the empty graph,} \\ 1 + \min_{v \in V(G)} \mathsf{td}(G - v) & \text{if } G \text{ is connected,} \\ \max\{\mathsf{td}(H) \mid H \text{ is a connected component of } G\} & \text{otherwise.} \end{cases}$$

In other words, at each step, we remove one vertex from each connected component of G, and the treedepth of G is the minimum number of steps necessary to remove all vertices. It can be seen as a measure on the connectivity of a graph: the easier a graph G can be broken in many small components, the smaller its treedepth is. Reidl, Rossmanith, Villaamil, and Sikdar [256] provide an algorithm computing the treedepth of a graph in time $2^{\mathcal{O}(\mathsf{td}^2)} \cdot n$. Papers studying the parameterization by td include [37, 162, 174, 256, 312].

[9] $\mathcal{H}$ is *closed under disjoint union* if the disjoint union of any two graphs in $\mathcal{H}$ is also in $\mathcal{H}$.

Elimination distance to $\mathcal{H}$. From this definition of treedepth, we may wonder: what if we stop when G belongs to some target class $\mathcal{H}$ instead of when G is the empty graph? In this case, we call it the *elimination distance* of G to $\mathcal{H}$, that was defined by Bulian and Dawar about 10 years ago [43, 44]. An equivalent way to define it is the following. The *elimination distance* of G to a graph class $\mathcal{H}$ is the minimum $k \in \mathbb{N}$ such that there exists a modulator X with $\mathsf{td}(\mathsf{torso}(G, X)) \leq k$ and each connected component of $G - X$ belongs to $\mathcal{H}$. Hence, the problem ELIMINATION DISTANCE TO $\mathcal{H}$, that asks, given a graph G and a $k \in \mathbb{N}$, whether G has elimination distance at most k, is a graph modification problem, where the target class is $\mathcal{H}$, the modification is vertex deletion, and the measure on the modulator is the treedepth of its torso. ELIMINATION DISTANCE TO $\mathcal{H}$ can be seen as some kind of "parallel" VERTEX DELETION TO $\mathcal{H}$ problem, where we delete vertices simultaneously on each connected component.

ELIMINATION DISTANCE TO $\mathcal{H}$ is solvable in FPT-time for any target class $\mathcal{H}$ such that $\mathsf{obs}_{\preceq_\mathsf{i}}(\mathcal{H})$ or $\mathsf{obs}_{\preceq_\mathsf{tm}}(\mathcal{H})$ is finite [6, 7], such that the graphs in $\mathcal{H}$ have bounded degree [7], such that $\mathcal{H}$ is the class of cliques [10], or such that $\mathcal{H}$ is minor-closed [44, 278]. Additionally, when the property of belonging to $\mathcal{H}$ is expressible by a FO formula φ (cf. Section 1.6 for the definition of FO logic), sufficient and necessary conditions on the prefix of φ are known for the existence of an FPT-algorithm for the problem [123].

This "torso idea" can be generalized to every graph parameter.

Torso-parameters

Given a graph parameter p and a graph class $\mathcal{H}$, we define $\mathcal{H}$-p to be the parameter mapping each graph G to

$$\mathcal{H}\text{-}\mathsf{p}(G) = \min\{k \in \mathbb{N} \mid \exists X \subseteq V(G), \mathsf{p}(\mathsf{torso}(G, X)) \leq k \text{ and the components of } G - X \text{ are in } \mathcal{H}\}.$$

If $\mathcal{H}$ is a graph class closed under disjoint union, then (G, k) is a yes-instance of VERTEX DELETION TO $\mathcal{H}$ if and only if $\mathcal{H}$-$\mathsf{size}(G) \leq k$, where size is the graph parameter that maps a graph to its number of vertices. When $\mathsf{p} = \mathsf{td}$, then $\mathcal{H}$-td is the *elimination distance* to $\mathcal{H}$. More generally, for any graph parameter p, $\mathcal{H}$-p corresponds to a different graph modification problem where the target class is $\mathcal{H}$, the modification is vertex deletion, and the measure is p applied to the torso of the modulator.

Those torso parameters are rather new and are thus not well-studied yet. In particular, we stick here to vertex deletion problems, as we are not aware of such a measure for another modification operation. Let us present a last well-known $\mathcal{H}$-p parameter.

$\mathcal{H}$-treewidth. When $\mathsf{p} = \mathsf{tw}$, then this is $\mathcal{H}$-tw, commonly known as *$\mathcal{H}$-treewidth*, which was defined by Eiben, Ganian, Hamm, and Kwon in [104]. Just as presented in the beginning of this part, their idea was to combine the best of treewidth and of the target class $\mathcal{H}$. Given an instance (G, k), checking whether $\mathcal{H}$-$\mathsf{tw}(G) \leq k$ can be done in FPT-time when $\mathcal{H}$ is bipartite, when $|\mathsf{obs}_{\preceq_i}(\mathcal{H})|$ is finite [173], when $\mathcal{H}$ has bounded rankwidth [104] or is minor-closed [205, 278].

FPT-equivalence of torso-parameters

Let us give a few more definitions. We say that two parameters p and p' are such that $\mathsf{p} \preceq \mathsf{p}'$ if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that, for any graph G, $\mathsf{p}(G) \leq f(\mathsf{p}'(G))$. Two parameters p and p' are *asymptotically equivalent*, denoted by $\mathsf{p} \sim \mathsf{p}'$ if $\mathsf{p} \preceq \mathsf{p}'$ and $\mathsf{p}' \preceq \mathsf{p}$. A graph parameter p is said to be *minor-monotone* if, for any graph G and any minor H of G, $\mathsf{p}(H) \leq \mathsf{p}(G)$. We call *Hadwiger* *number* of a graph G, denoted by $\mathsf{hw}(G)$, the maximum $t \in \mathbb{N}$ such that K_t is a minor of G. Observe that, for any (unbounded) minor-monotone parameter p, we have

$$\mathsf{hw} \preceq \mathsf{p} \preceq \mathsf{size}.$$

An important result on those $\mathcal{H}$-p parameters was proved by Agrawal, Kanesh, Lokshtanov, Panolan, Ramanujan, Saurabh, and Zehavi in [6]: given a parameter $\mathsf{p} \in \{\mathsf{size}, \mathsf{td}, \mathsf{tw}\}$ and a target class $\mathcal{H}$ that is hereditary and CMSO-definable (see Section 1.6 for the definition), if checking whether $\mathcal{H}$-$\mathsf{p}(G) \leq k$ is in FPT, then the same holds for the two other parameters. More importantly, the techniques used in [6] seem applicable to any minor-monotone parameter p with $\mathsf{tw} \preceq \mathsf{p} \preceq \mathsf{size}$. It would essentially imply that it is enough to study graph modification problems whose measure is the size of the modulator, instead of other torso measures (for $\mathsf{tw} \preceq \mathsf{p} \preceq \mathsf{size}$). However, their methods are not applicable to parameters p with $\mathsf{hw} \preceq \mathsf{p} \prec \mathsf{tw}$.

Hence, one of our objectives is the following.

> Given graph class $\mathcal{H}$ and a graph parameter p with $\mathsf{hw} \preceq \mathsf{p} \prec \mathsf{tw}$, study the problem of checking, given a graph G and a $k \in \mathbb{N}$, whether $\mathcal{H}$-$\mathsf{p}(G) \leq k$.

We do so in Chapter 10, whose results are presented in Section 2.6.

1.6 Logic

In Section 1.3, Section 1.4, and Section 1.5, we presented ways to express many graph modification problems at once, by encompassing several target classes, or several modifications, or several measures on the modulator. Still there is one way that we did not mention yet, that encompasses those all at once. This way is logic.

While logic is not the main focus of this thesis, it is necessary to present some of the most used logics in Graph Theory, in order to introduce the algorithmic meta-theorems that can be applied to graph modification problems. Let us in particular define FO logic and CMSO logic.

First-Order Logic. The syntax of *First-Order Logic* (FO) includes logical connectives $\wedge$, $\vee$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices and edges, quantifiers $\exists$, $\forall$ over these variables, and the relations $\mathrm{adj}(u,v)$ when u and v are vertex variables, with the interpretation that u and v are adjacent, and $\mathrm{inc}(u,e)$ when u is a vertex variable and e is an edge variable, with the interpretation that e is incident to u; and equality of variables representing vertices and edges. For instance, the formula $\exists v_1, \exists v_2, \forall e, (\mathrm{inc}(v_1,e) \vee \mathrm{inc}(v_2,e))$ expresses the property of having a vertex cover of size two, that is, that there exists two vertices v_1 and v_2 such that any edge e has v_1 or v_2 as an endpoint. More generally, FO logic can express problems such as k-Vertex Cover, k-Dominating Set, or (Induced) Subgraph Containment. Concerning graph modification problems, the authors of [122] studied the problems of removing/adding at most k vertices/edges when the target class is definable by a first-order formula φ. They gave sufficient and necessary conditions on the structure of the prefix of φ specifying when the corresponding graph modification problem is FPT (parameterized by k) and when it admits a polynomial kernel.

(Counting) Monadic Second-Order Logic. The syntax of *Monadic Second-Order Logic* (MSO) extends FO by including variables for vertex sets and edge sets as well as the atomic expressions

$u \in V$ when u is a vertex variable and U is a vertex set variable and $e \in E$ when e is a edge variable and E is a edge set variable. MSO logic can express, aside from all problems expressible in FO logic, problems such as k-Colorability or Connectivity. *Counting Monadic Second-Order Logic* (CMSO) extends MSO by including atomic sentences testing whether the cardinality of a set is equal to q modulo r, where $r \in \mathbb{N}_{\geq 2}$ and $q \in [r-1]$. We say that a class of graphs $\mathcal{H}$ is *CMSO-definable* if the property of belonging to $\mathcal{H}$ is expressible in CMSO logic.

Model-checking

Let us see how logic can be employed to state algorithmic meta-theorems.

The Model-Checking problem asks, given a formula φ and a graph G, whether G satisfies φ. Many algorithmic meta-theorems are stated as follows: *Model-checking for a certain logic is solvable in a certain time for graphs of a certain graph class.* The most famous such meta-theorem is Courcelle's theorem [18, 67] that states that Model-Checking for a formula φ definable in CMSO logic is solvable on graphs of treewidth at most tw in time $f(\mathsf{tw}, |\varphi|) \cdot n$ for some computable function f. When the expressibility of the logic is restricted, then one may find tractable results for broader input graph classes. As such, Model-Checking for a formula φ definable in FO logic is solvable in FPT-time on nowhere-dense graphs [153] or on bounded twin-width graphs [41]. Between FO and CMSO logic, other logics were defined, each with a corresponding meta-theorem that states on which graph class Model-Checking this logic is FPT. We refer to [152, 208, 295] for surveys on the subject. Many graph modification problems can be expressed in some logic. For instance, if $\mathcal{H}$ is a graph class definable in CMSO logic (as is often the case), then so is k-Vertex Deletion to $\mathcal{H}$. As such, k-Vertex Deletion to $\mathcal{H}$ is solvable in linear FPT-time on graphs of bounded treewidth by Courcelle's theorem.

Logics for graph modification problems

Some logics were specifically created to express graph modification problems.

Fomin, Golovach, Sau, Stamoulis, and Thilikos developed in [120] such a logic, the Θ-logic, for which Model-Checking can be done in quadratic FPT-time. This logic is based on the modulator versus target scheme. In particular, it can express graph modification problems where:

- the torso of the *modulator* X has treewidth at most k and $G[X]$ or $\mathsf{torso}(G, X)$ belongs to a CMSO-definable graph class, and
- $G - X$, or its connected components, belongs to a *target* graph class $\mathcal{H}$ expressible by a positive boolean combination of sentences of the form $\sigma \wedge \mu$, where σ is a FO-sentence and μ expresses non-trivial minor-exclusion.

Most of the studied graph modification problems where the target class $\mathcal{H}$ is minor-closed or FO-definable fit into this Θ-logic. In particular, this is the case of the problem of checking, given a graph G and a $k \in \mathbb{N}$, whether $\mathcal{H}$-$\mathsf{p}(G) \leq k$, for any such target class $\mathcal{H}$ and any graph parameter p with $\mathsf{p} \preceq \mathsf{tw}$ (not necessarily minor-closed) that can be expressed in CMSO logic.

The technique employed to construct the algorithm of [120] is based on the irrelevant vertex technique developed by Robertson and Seymour in their Graph Minor series [271] (see Subsection 3.1.3 for more on the subject), that is further developed here to be applicable to many graph modification problems. A subset of the authors of [120] later tried to push the irrelevant vertex technique to its very limit and created an algorithmic meta-theorem in [287] that encompasses the results of [120]. To present their new logic, we first need to define annotated parameters.

Annotated parameters. Given two graphs G and H and a set $X \subseteq V(G)$, we say that H is an *X-rooted minor* of G (or simply an *X-minor* of G) if there is a collection $\mathcal{S} = \{S_v \mid v \in V(H)\}$ of pairwise-disjoint connected[10] subsets of $V(G)$, each containing at least one vertex of X and such that, for every edge $xy \in E(H)$, the set $S_x \cup S_y$ is connected in G. An *annotated graph* is a pair (G, X) where G is a graph and $X \subseteq V(G)$. The annotated version of a graph parameter p is defined as follows: for each annotated graph (G, X),

$$\mathsf{p}(G, X) = \max\{\mathsf{p}(H) \mid H \text{ is a } X\text{-rooted minor of } G\}.$$

Note that the annotated size of a set X is equal to its size, and that $\mathsf{p}(G, V(G)) = \mathsf{p}(G)$ for each graph G. Also, for each graph G and each $X \subseteq V(G)$, $\mathsf{tw}(G, X) \le \mathsf{tw}(\mathsf{torso}(G, X))$ [302].

Sau, Stamoulis, and Thilikos introduce in [287] the CMSO/tw logic. It extends CMSO logic by replacing the quantifier $\exists$ by the quantifier $\exists_{\mathsf{tw} \le k} := \exists X(\mathsf{tw}(G, X) \le k)$. They prove that Model-Checking, given a formula φ in CMSO/tw and a graph G, is solvable in time $f(|\varphi|, \mathsf{hw}(G)) \cdot n^2$. This generalizes all algorithmic meta-theorems described above in the context of graph modification problems, where the target class is restricted to being minor-closed. Additionally, it implies that, for any graph modification problem Π where the target class $\mathcal{H}$ is minor-closed, where the modulator is quantified by some parameter p larger than tw, and the modification is expressible in CMSO logic, Π is solvable in quadratic FPT-time.

In all the algorithmic meta-theorems presented above, the parametric dependence is not explicit, because it is typically huge. From this stems one of the main goals of this thesis:

> Prove algorithmic meta-theorems for graph modification problems to minor-closed graph classes with an explicit parametric dependence.

This is what we do in Part III, whose results are described in Section 2.2, Section 2.3, and Section 2.4.

Bidimensionality

Let us discuss about what the annotated treewidth $\mathsf{tw}(G, X)$ represents. The *biggest grid* bg is the maximum k such that G contains a $(k \times k)$-grid as a minor (see Figure 1.19). A $(k \times k)$-grid has

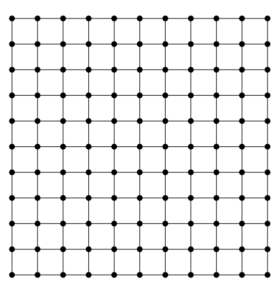

Figure 1.19: A (11×11)-grid.

treewidth k, implying that $\mathsf{bg} \preceq \mathsf{tw}$. A fundamental structural result is the celebrated “grid exclusion theorem” proved by Robertson and Seymour [261] for the opposite direction. It states the existence of a function $f : \mathbb{N} \to \mathbb{N}$ such that, for every graph G, $\mathsf{tw}(G) \le f(\mathsf{bg}(G))$, and thus implies that

[10] A set X is *connected* in G if the subgraph of G induced by X is a connected graph.

$\mathsf{tw} \sim \mathsf{bg}$. Chekuri and Chuzhoy in [56] prove that the function f can be chosen to be polynomial, and the current best upper bound is $f(k) = k^9 \cdot \log^{\mathcal{O}(1)} k$ by Chuzhoy and Tan in [65].

The *bidimensionality* of a vertex set X of G is defined as $\mathsf{bg}(G, X)$, that is, the maximum $\mathsf{bg}(H)$ over all X-minors H of G. Intuitively, bidimensionality measures to what extent X can be "spread" as a 2-dimensional grid inside the graph G. The irrelevant vertex technique that is used to prove the result of [287] and most of the results on graph modification problems to minor-closed graph classes actually relies in particular on the existence of a big enough grid Γ as a subgraph in G that avoids the modulator X. If the bidimensionality of X, and its annotated treewidth, is bounded, then such a Γ can always be found in a bigger grid that is a minor of G (possibly containing vertices of X). If the annotated treewidth of X is unbounded however, then we might not be able to find such a Γ, and thus we cannot use the irrelevant vertex technique in this case. This explains why the meta-theorems of [120, 287] only work for graph modification problems where the modulator X has bounded bidimensionality (i.e., such that $\mathsf{tw}(\mathsf{torso}(G, X))$ or $\mathsf{tw}(G, X)$ is bounded).

One of the motivations of this thesis is thus the following:

> Develop new techniques to solve graph modification problems where the modulator has unbounded bidimensionality.

See Section 2.6 for a description of our results on this subject and Chapter 10 for the proof of those results.

While the algorithmic meta-theorem of [287] pushes the irrelevant vertex technique to its very limit, its authors do not provide any studied graph modification problem expressible by CMSO/tw logic but not in the Θ-logic of [120]. There actually exist such results with modifications of bounded bidimensionality, but, for this, we need to turn our attention towards structure theorems.

1.7 Structure theorems

Graph structure theorems describe the structure of graphs with some property, which often corresponds to the exclusion of some graph H in some way. Interestingly, some structure theorems can be expressed using terminology from graph modification problems.

1.7.1 Excluding a graph as a minor

Grid exclusion theorem

The "grid exclusion theorem", proved by Robertson and Seymour in [261], asserts that, for every *planar* graph H, there is a constant c_H such that every graph excluding H as a minor has a *tree decomposition* of width most c_H (see also Section 1.6).

The grid exclusion theorem implies win/win strategies to solve graph modification problems. A first example is when the target class is minor-closed. If the treewidth of the input graph G is big, then G contains a big grid as a minor, to which we can apply the irrelevant vertex technique that is mentioned in Section 1.6 and is detailed more in Chapter 3 to reduce the size of G. Otherwise, the treewidth of G is small, in which case we may apply Courcelle's theorem or use a dynamic programming approach to conclude. A second example follows from the *theory of bidimensionality* [81], which examines various consequences of grid-containment in graph algorithms. Here, a big grid minor usually translates into an immediate no (or yes) answer, while one may, for instance, solve the problem in subexponential time (in the parameter) otherwise. Other applications of bidimensionality theory concern EPTAS [83] and kernelization [129].

Graph minors structure theorem

An important question is whether a "similar" tree decomposition theorem exists when the excluded graph is a non-planar graph. Just as is the case for the grid exclusion theorem, knowing such a tree-like decomposition for a graph G could imply efficient algorithms to solve some problems on G. A general answer to this question was provided by the celebrated *graph minors structure theorem* (GMST) of Robertson and Seymour [275] asserting that graphs excluding a (non-necessarily planar) graph H as a minor can be tree-decomposed such that the torso at each node is "c_H-almost embeddable" after removing at most c_H vertices called *apices*, where c_H is a constant depending of $|V(H)|$ (see Figure 1.20). Intuitively, a graph G is *k-almost embeddable* if G can be drawn in a surface Σ of Euler genus at most k such that crossings only happen in at most k regions called *vortices* of "width" at most k. The *width* of a vortex essentially corresponds to its pathwidth[11]. Let us mention that Gorsky, Seweryn, and Wiederrecht very recently proved that the function mapping H to c_H is polynomial [148], while it was immense in [275]. See also [84] for some applications of the GMST.

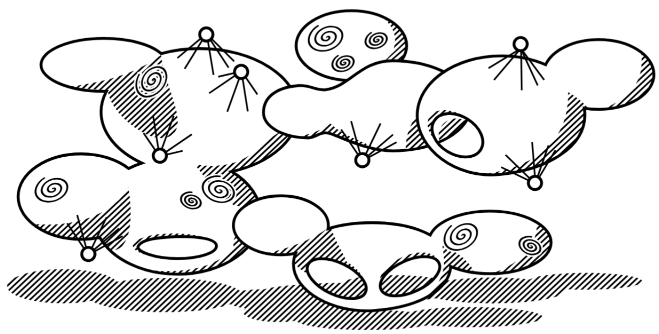

Figure 1.20: Artistic illustration by Felix Reidl of the structure of a graph excluding another graph as a minor. Vortices are represented by spirals and the apices are the points above the surfaces.

The GMST is one of the cornerstone results of the Graph Minors series of Robertson and Seymour (see [194, 195] for a relatively simpler and self-contained proof). This result, being general enough, does not provide refined enough structure when the excluded graphs enjoy particular structural properties. The general research program of proving refined versions of GMST can be outlined as follows.

> *Given a class of graphs $\mathcal{H}$, find a graph parameter, defined in terms of tree decompositions, such that its value is bounded for graphs excluding some graph in $\mathcal{H}$ as a minor and, moreover, its value is unbounded for the graphs in $\mathcal{H}$.* (1.1)

Clearly the GMST, being a general and "all purpose" theorem, provides only a partial answer to Question (1.1) and it is a challenge to detect which tree decomposition parameter is the suitable answer for certain instantiations of the class $\mathcal{H}$. As we already mentioned, when $\mathcal{H}$ is the class of planar graphs, denoted by $\mathcal{G}_{\mathsf{planar}}$, the parameter in Question (1.1) can be chosen to be the parameter of treewidth (or any other parameter that is equivalent to treewidth).

[11] A *path decomposition* of a graph G is a tree decomposition (T, β) such that T is a path, and the *pathwidth* of G is the smallest width of a path decomposition of G.

Clique-sum extension of parameters

To further formalize Question (1.1), we introduce a general framework based on tree decompositions. The *clique-sum closure* of a graph class $\mathcal{G}$, denoted by $\mathcal{G}^\star$, is the graph class containing every graph that has a tree decomposition such that the torso at each node belongs in $\mathcal{G}$. One may use tree decompositions in order to define graph parameters using simpler ones as follows: if $\mathsf{p}\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ is a graph parameter, where $\mathcal{G}_{\mathsf{all}}$ is the class of all graphs, then we define the *clique-sum extension* of p as the graph parameter $\mathsf{p}^\star\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ such that

$$\mathsf{p}^\star(G) = \min\left\{k \mid G \in \{H \mid \mathsf{p}(H) \leq k\}^\star\right\},$$

in other words, $\mathsf{p}^\star(G) \leq k$ if and only if G is in the clique-sum closure of the graphs where the value of p is at most k. Using this notation and given some graph class $\mathcal{H}$, we may formalize Question (1.1) as follows:

Find a minor-monotone parameter $\mathsf{p}_{\mathcal{H}}\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$, *such that*

(A) there is a function $f\colon \mathbb{N} \to \mathbb{N}$ *where for every* $H \in \mathcal{H}$, *if a graph* G *excludes the graph* H *as a minor then* $\mathsf{p}^\star_{\mathcal{H}}(G) \leq f(|H|)$, *and* (1.2)

(B) the values of $\mathsf{p}^\star_{\mathcal{H}}$ *for the graphs in* $\mathcal{H}$ *are unbounded.*

For the case of the class $\mathcal{G}_{\mathsf{planar}}$ of planar graphs, we may pick $\mathsf{p}_{\mathcal{G}_{\mathsf{planar}}} := \mathsf{size}$, where $\mathsf{size}\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ maps a graph G to its size $|V(G)|$. Indeed, it is enough to observe that the treewidth of a graph G is equal to $\mathsf{size}^\star(G) - 1$.

In the most general case where $\mathcal{H} = \mathcal{G}_{\mathsf{all}}$, an answer to Question (1.2) is given by the GMST. Here, we may pick $\mathsf{p}_{\mathcal{G}_{\mathsf{all}}}$ as the parameter mapping a graph G to the minimum k for which G contains at most k apices whose removal yields a graph that is k-almost embeddable.

Excluding an almost planar graph

Researchers studied Question (1.2) for graph classes $\mathcal{H}$ that are "close to be planar". In this direction, Robertson and Seymour consider in [268] the class of *singly-crossing graphs*, i.e., graphs that can be drawn in the sphere such that there is at most one pair of edges that share a common point. For this, we define $\mathcal{G}_{\mathsf{singly\text{-}crossing}}$ as the class of all minors of singly-crossing graphs. According to [268], if G excludes a singly-crossing graph H as a minor, then G has a tree decomposition whose non-planar torsos have size that is bounded by some constant, depending on H. Assume now that $\mathsf{psize}\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ is the graph parameter where $\mathsf{psize}(G)$ is zero if G is planar, and otherwise $\mathsf{psize}(G)$ is the size of G. It can be proved that the values of $\mathsf{psize}^\star$ are unbounded for the graphs in $\mathcal{G}_{\mathsf{singly\text{-}crossing}}$. This, along with the the aforementioned result of [268] implies that we may pick $\mathsf{p}_{\mathcal{G}_{\mathsf{singly\text{-}crossing}}} := \mathsf{psize}$ so as to provide an answer to Question (1.2) when $\mathcal{H} = \mathcal{G}_{\mathsf{singly\text{-}crossing}}$.

More recently, Dvořák and Thomas consider in [101] the class $\mathcal{G}_{t\text{-}\mathsf{apex}}$ of *t-apex graphs*, i.e., graphs that are planar after removing t vertices (an apex graph is exactly a 1-apex graph). They prove that, if G excludes a t-apex graph H as a minor, then $\mathsf{p}^*_{\mathcal{G}_{t\text{-}\mathsf{apex}}}(G) \leq c_H$ for some constant c_H depending on $|V(H)|$, where $\mathsf{p}_{\mathcal{G}_{t\text{-}\mathsf{apex}}}$ is the parameter mapping each graph F to the minimum k for which

- F contains at most k apices whose removal yields a graph that is k-almost embeddable and
- such that all apices but at most t are only adjacent to other apices and vertices in vortices.

Bounded bidimensionality modulator

Thilikos and Wiederrecht [302] restate the GMST using a *modification operation*. They prove that $\mathsf{p}_{\mathcal{G}_{\mathsf{all}}}$ can also be picked to be the parameter mapping each graph G to the minimum k such that there is a set $X \subseteq V(G)$ of *bidimensionality* at most k such that $G - X$ is embeddable in a surface of genus at most k. So the modification is stated in terms of *vertex removals* and the *target* of this modification is surface embeddability. More refined results have been proved in [302] providing an answer to Question (1.2) in the case where $\mathcal{H}$ is the class of graphs embeddable in some particular surface. For every choice of a surface, the resulting graph parameter is defined in terms of modifications based on vertex deletion.

Hence, one of our goals is the following.

> Find a structure theorem for the exclusion of a graph class $\mathcal{G}$ that is "close to be planar" such that $\mathsf{p}_{\mathcal{G}}$ can be expressed in terms of modifications.

See Section 2.1 and Chapter 5 for our structural results related to the above question.

1.7.2 Excluding a graph as an odd-minor

Structure theorems also exist for partial orders on graphs other than minor containment. Given that we explain in Subsection 1.3.2 that odd-minor-closed graph classes might be as promising as minor-closed graph classes, let us study structure theorems for odd-minors more in details.

Demaine, Hajiaghayi, and Kawarabayashi prove in [85] that, if G excludes a graph H as an *odd-minor*, then $\mathsf{p}^*_{\mathsf{odd},\mathcal{G}_{\mathsf{all}}}(G) \leq c_H$ for some constant c_H depending on $|V(H)|$, where $\mathsf{p}_{\mathsf{odd},\mathcal{G}_{\mathsf{all}}}$ is the parameter mapping each graph F to the minimum k for which F contains at most c_H apices whose removal yields either

- a bipartite graph or
- a c_H-almost-embeddable graph.

Similar results exist for other containment relations on graphs [102, 154, 185].

Bipartite treewidth modulator

Tazari observe in [299] from the proof of [85] that the structure theorem for odd-minors can be alternatively stated as follows[12]: if G excludes a graph H as an *odd-minor*, then $\mathcal{H}_H\text{-}\mathsf{btw}(G) \leq c_H$, where $\mathcal{H}_H$ is the class of graphs excluding K_{d_H,e_H} as a minor, for some constants c_H, d_H, and e_H depending on $|V(H)|$. Here, the *bipartite treewidth* btw of a graph G is the minimum width of a bipartite tree decomposition of G, which is a tree decomposition such that the bag of each node induces a bipartite graph after the removal of at most k vertices and such that the intersection of two bags contains at most one vertex of the bipartite part of each side (see Figure 1.21 for an illustration).

Bipartite treewidth is used implicitly by Kawarabayashi and Reed [192] in order to solve Odd Cycle Transversal parameterized by the solution size. Campbell, Gollin, Hendrey, and Wiederrecht [49] are also currently studying bipartite tree decompositions. In particular, they provide universal obstructions characterizing bounded btw in the form of a "grid exclusion theorem" (actually

[12] Our statement is a bit different from the one in [299], but can easily be deduced from [85, 299].

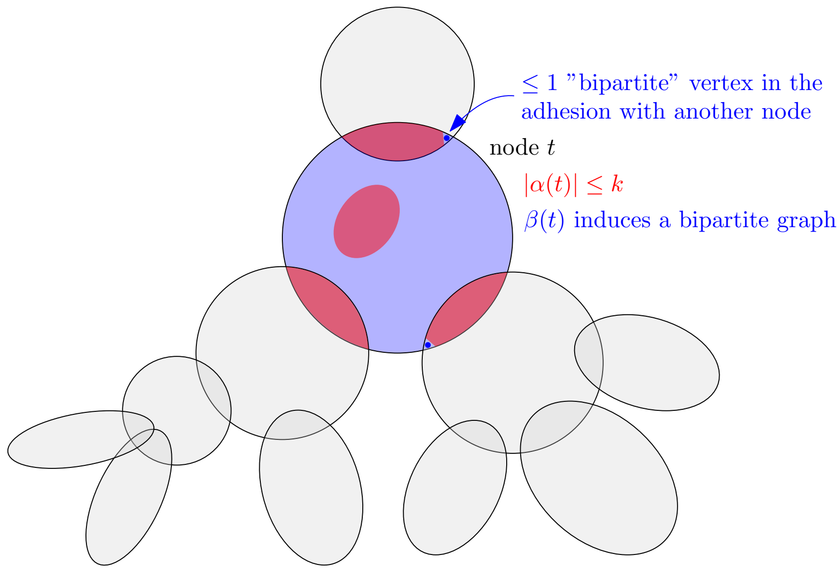

Figure 1.21: Illustration of a bipartite tree decomposition. The blue part induces a bipartite graph, while the additional vertices are depicted in red. This bipartite tree decomposition has width at most k if there are at most k additional vertices in each bag.

the result of [49] applies in the much more general setting of undirected group labeled graphs). They also design an algorithm that either constructs a bipartite tree decomposition of the input graph G of width at most $f(k)$ in time $g(k) \cdot n^4 \log n$ for some computable functions f, g, or reports that $\mathsf{btw}(G) > k$.

From this characterization of graph classes excluding an odd-minor, it appears that a first step to be able to solve problems on odd-minor-closed graph classes is to be able to solve the corresponding problems on graphs of bounded bipartite treewidth. Hence, one of our objective is the following.

Solve (graph modification) problems parameterized by bipartite treewidth.

See Section 2.5 and Chapter 9 for our results on the subject.

Organization of the thesis. The thesis is organized as follows. The results of the thesis are presented in Chapter 2. The main techniques used throughout the thesis are discussed in Chapter 3. In Chapter 4, we give some preliminary definitions and results. Chapter 5, Chapter 6, Chapter 7, Chapter 8, Chapter 9, and Chapter 10 are dedicated to proving the results of this thesis, and are presented more in detail in Chapter 2. Finally, concluding remarks and research directions are available in Chapter 11.

CHAPTER 2

Results

Contents

In this chapter, we present the results of this thesis.

In Part II of the thesis, we give a new example of the use of bounded bidimensionality modulators via a structure theorem, which is presented in Section 2.1.

In Part III, we design FPT-algorithms with a better running time for many graph modification problems that are already solvable using algorithmic meta-theorems, in particular improving the parametric dependence. Those results towards efficiency are presented in Section 2.2, Section 2.3, and Section 2.4.

In Part IV, we develop new techniques to solve graph modification problems beyond the scope of current algorithmic meta-theorems. Those results towards generalization are presented in Section 2.5 and Section 2.6.

Finally, the papers written during this thesis are listed in Section 2.7.

2.1 Excluding edge-apex graphs

In Subsection 1.7.1, we mention that the GMST can be expressed in terms of deletion of a set of bounded bidimensionality [302]. We present here a new structure theorem that can also be expressed in terms of modification of a modulator of bounded bidimensionality, though the modification is not vertex deletion anymore. This structure theorem concerns the exclusion of graphs that are close to planar in yet another sense than singly-crossing graphs or apex graphs.

An *edge-apex graph* is a graph G containing an edge e such that $G - e$ is planar. Note that we have the inclusion

$$\text{singly-crossing graphs} \subseteq \text{edge-apex graphs} \subseteq \text{apex graphs}.$$

Therefore, if $\mathcal{G}_{\mathsf{edge\text{-}apex}}$ is the class of edge-apex graphs, then the corresponding parameter $\mathsf{p}_{\mathcal{G}_{\mathsf{edge\text{-}apex}}}$ (cf. Subsection 1.7.1) should be such that $\mathsf{p}_{\mathcal{G}_{t\text{-}\mathsf{apex}}} \preceq \mathsf{p}_{\mathcal{G}_{\mathsf{edge\text{-}apex}}} \preceq \mathsf{p}_{\mathcal{G}_{\mathsf{singly\text{-}crossing}}}$, i.e. the tree decomposition when excluding an edge-apex graph should be more refined than the one when excluding an apex graph.

Let us give an equivalent definition of $\mathcal{G}_{\mathsf{edge\text{-}apex}}$ via pinched surfaces.

Pinched surfaces. Given a surface Σ, the *pinched* version of Σ, denoted by Σ°, is the pseudo-surface obtained if we identify two distinct points of Σ. According to Knor [199], for every surface Σ, the class of graphs embeddable in Σ° is minor-closed. Clearly, one may define pseudo-surfaces by identifying more sets of points of surfaces or unions of surfaces. However, as proved in [199], minor-closedness of graphs embeddable in pseudo-surfaces holds only for the pinched surfaces and for the pseudo-surfaces that are "spherically reducible" to them (see also [293]).

Equivalently, $\mathcal{G}_{\mathsf{edge\text{-}apex}}$ is the class of graphs embeddable in the pseudo-surface $\mathbb{S}_0^\circ$ obtained from the sphere $\mathbb{S}_0$ by identifying two points of $\mathbb{S}_0$, that we call the *pinched sphere*. Clearly the pinched sphere $\mathbb{S}_0^\circ$ can be seen as the set of points of the horn-torus

$$\{(x, y, z) \in \mathbb{R}^3 \mid (x^2 + y^2 + z^2)^2 = 4(x^2 + y^2)\}$$

depicted in Figure 2.1.

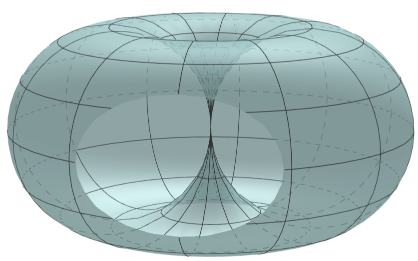

Figure 2.1: The pinched sphere with a hole, permitting the visibility of the pinch point (picture by Dimitrios M. Thilikos).

We first prove the following (see also Theorem 5.2.36).

Theorem 2.1.1. *Let H be an edge-apex graph. Then there is a constant c_H such that, if a graph G is H-minor-free, then $\mathsf{p}^\star(G) \leq c_H$, where p is the parameter mapping each graph F to the minimum k such that F is k-almost embeddable in the projective plane.*

In other words, an edge-apex minor-free graph has a tree decomposition such that the torso of each bag can be drawn in the projective plane with only a few vortices of small width. Interestingly, to our knowledge, this is the first structure theorem with *vortices* but no *apices*. Additionally, the constant c_H is essentially a function of the minimum number of crossings of the edge e in the planar embedding of $G - e$ (the "length of the jump").

Given that the GMST can be restated in terms of deletion of a vertex set of bounded bidimensionality to a surface [302], one may suspect that Theorem 2.1.1 can also be restated with some modification-based parameter where the modification operation is vertex deletion. As we show, it is correct that we have to consider some modification-based parameter, however vertex deletion does not work in this case. Instead, we need a modification operation based on *vertex identifications*. To formalize this, we need some more definitions. Given a graph G and a set $X \subseteq V(G)$, we set $\mathcal{I}(G, X)$ to be the set of all graphs G' obtained from G by identifying each part X_i of a partition $(X_1, \ldots, X_r)$ of X to a single vertex x_i, $i \in [r]$. Let $\mathcal{G}_{\mathsf{projective}}$ be the class of graphs embeddable in the projective plane. We define $\mathsf{idpr}\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ as the graph parameter where

$$\mathsf{idpr}(G) \ := \ \min\{k \mid \exists X \subseteq V(G),\ \mathsf{bg}(G, X) \leq k \text{ and } \mathcal{I}(G, X) \cap \mathcal{G}_{\mathsf{projective}} \neq \emptyset\}, \tag{2.1}$$

in other words, $\mathsf{idpr}(G) \leq k$ if and only if G contains a set X of bidimensionality at most k such that a projective graph can be obtained by identifying vertices of X in G.

Then we prove the following (see also Theorem 5.2.39), which corresponds to item (A) of Question (1.2) for edge-apex graphs.

Theorem 2.1.2. *Let H be an edge-apex graph. There is a constant c_H such that, if a graph G is H-minor-free, then $\mathsf{idpr}^\star(G) \leq c_H$.*

Moreover, this result is “parametrically tight”, in the sense of Question (1.2), given that we prove the following (see also Theorem 5.3.8), which corresponds to item (B) of Question (1.2).

Theorem 2.1.3. *For any $h \in \mathbb{N}$, there is an edge-apex graph H such that $\mathsf{idpr}^\star(H) \geq h$.*

Identification versus contraction. Actually, Theorem 2.1.2 could have been stated using contractions instead of identifications (with a bit more work). However, we crucially use the fact that idpr is a minor-monotone parameter to prove Theorem 2.1.3, which is not the case anymore if we contract edges instead of identifying vertices, as discussed in Section 6.4.

Suppose now that we define a variant of idpr, namely idpl by considering in (2.1) $\mathcal{G}_{\mathsf{planar}}$ instead of $\mathcal{G}_{\mathsf{projective}}$, i.e., we now demand that the surfaces where the torsos of the tree decomposition are almost embedded are spheres. By a simple variant of our proof strategy, we obtain the following variant of Theorem 2.1.2 (see also Theorem 5.2.40).

Theorem 2.1.4. *For every $H_1 \in \mathcal{G}_{\mathsf{projective}}$ and every $H_2 \in \mathcal{G}_{\mathsf{edge\text{-}apex}}$, there exist a constant c_{H_1,H_2} such that, if a graph G excludes both H_1 and H_2 as a minor, then $\mathsf{idpl}^\star(G) \leq c_{H_1,H_2}$.*

The optimality of the decomposition of Theorem 2.1.4 for the set of classes $\{\mathcal{G}_{\mathsf{projective}}, \mathcal{G}_{\mathsf{edge\text{-}apex}}\}$ follows by Theorem 2.1.3 and its counterpart for $\mathcal{G}_{\mathsf{projective}}$ and $\mathsf{idpl}^\star$, shown in Theorem 5.3.9.

All those results are proved in Chapter 5.

2.2 Identification to a forest

To our knowledge, graph modification problems where the modification is vertex identification have yet to be studied. The closest study is the one of [66] on vertex fusion, where the goal is to find a vertex set X of small size that can be identified to a single vertex x such that the remaining graph is in the target class. While close, this modification does not correspond to our structure theorem (cf. Section 2.1). Also, although we will not dwell on the subject, those familiar with *quotient graphs* or *homomorphisms* may observe the following equivalence: G admits an identification to $\mathcal{H}$ if and only if G admits a quotient graph that belongs to $\mathcal{H}$ if and only if there is a surjective homomorphism from G to a graph in $\mathcal{H}$ (sometimes called $\mathcal{H}$-coloring [31]). However, we are not aware of any optimization version of graph homomorphism (or graph quotient) to a fixed graph class that would fit our setting. Hence, after intuiting Theorem 2.1.2, we decided to tackle the problem.

We begin with the simplest measure on the modulator, that is the size of the modulator, instead of its bidimensionality. We define Identification to $\mathcal{H}$ as follows:

Identification to $\mathcal{H}$

Input:	A graph G, a $k \in \mathbb{N}$.
Question:	Is there a set $X \subseteq V(G)$ of size at most k such that $\mathcal{I}(G, X) \cap \mathcal{H} \neq \emptyset$?

When $\mathcal{H}$ is the class of edgeless graphs, (G, k) is trivially a yes-instance of Identification to $\mathcal{H}$ if and only if there are at least $n - k$ isolated vertices (each connected component C containing some edge needs to be identified to a single vertex v_C). When the target class is the class of forests $\mathcal{F}$, the problem is not as trivial.

A problem that is similar to Identification to $\mathcal{F}$ is Contraction to $\mathcal{F}$, asking whether it is possible to *contract* k edges in a graph G so to obtain an acyclic graph. According to the results by Heggernes, van 't Hof, Lokshtanov, and Paul in [165], this problem can be solved in time $4.98^k \cdot |G|^{\mathcal{O}(1)}$. As edge contractions are special cases of vertex identifications, if (G, k) is a yes-instance of Contraction to $\mathcal{F}$ then $(G, 2k)$ is also a yes-instance of Identification to $\mathcal{F}$. However, vertex identifications may not be edge contractions, and it is certainly possible that a yes-instance of Identification to $\mathcal{F}$ is certified by the identifications of non-adjacent vertices that cannot be simulated by a small number of edge contractions.

We prove the following hardness and parameterized results from a reduction to Vertex Cover (see also Theorem 6.1.7 and Theorem 6.1.10 respectively).

Theorem 2.2.1. Identification to Forest *is* NP*-complete.*

Theorem 2.2.2. *There is an algorithm that, given an instance (G, k) of* Identification to Forest*, outputs in time $\mathcal{O}(k\sqrt{\log k} \cdot n + k^3)$ an equivalent instance (G', k') where $|V(G')| \leq 2k + 1$ and $k' \leq k+1$. Alternatively, one can solve* Identification to Forest *in time $\mathcal{O}(1.2738^k + k\sqrt{\log k} \cdot n)$.*

Given $k \in \mathbb{N}$, the class $\mathcal{F}^{(k)}$ of graphs such that (G, k) is a yes-instance of Identification to Forest is a minor-closed graph class. Therefore, $\mathcal{F}^{(k)}$ has a finite set of (minor-)obstructions, that we can try to compute. We actually prove that the obstructions of $\mathcal{F}^{(k)}$ can be obtained from the obstructions of Vertex Cover by adding edges, and deduce the following (see also Theorem 6.2.13).

Theorem 2.2.3. *Let $k \in \mathbb{N}$. The obstructions of $\mathcal{F}^{(k)}$ have at most $2k + 4$ vertices.*

A linear upper bound as the above is known for the obstructions of the class $\mathcal{V}_k$ of graphs such that (G, k) is a yes-instance of Vertex Cover: Dinneen and Lai proved that $2k + 2$ is an upper bound on the size of the graphs in $\mathsf{obs}(\mathcal{V}_k)$ [90, 92].

All these results, plus some more about "identification-minors" and WQO, are available in Chapter 6.

2.3 Bounded size modulators to minor-closedness

After this study on identification to forests, we turned our attention towards identification to any minor-closed graph class, still parameterized by the size of the modulator.

Given that the yes-instances of Identification to $\mathcal{H}$ form a minor-closed graph class when $\mathcal{H}$ is minor-closed, we know from [205, 278] that there *exists* (non-constructively) an algorithm in time $\mathcal{O}_{s_\mathcal{H}}(f(k) \cdot n^{1+o(1)})$. By the algorithm meta-theorems of [120, 287], we can actually construct an algorithm solving Identification to $\mathcal{H}$ in time $\mathcal{O}_{s_\mathcal{H}}(f(k) \cdot n^2)$ for some computable function f. Now, our goal is to construct such an algorithm with the best (explicit) parametric dependence.

Prior to this thesis, the only results regarding bounded size modulators to minor-closedness (with an explicit parametric dependence) is the result of Sau, Stamoulis, and Thilikos [284] for Vertex Deletion to $\mathcal{H}$, with a running time of $2^{k^{\mathcal{O}_{s_\mathcal{H}}(1)}} \cdot n^3$, where $s_\mathcal{H}$ is the maximum size of an obstruction of $\mathcal{H}$, which is any minor-closed graph class. We actually first improved the algorithm for Vertex Deletion to $\mathcal{H}$ from a cubic to a quadratic time in [235]. However, when we later tried to solve Identification to $\mathcal{H}$ for any minor-closed graph class $\mathcal{H}$, we observed that the techniques used to solve Vertex Deletion to $\mathcal{H}$ can be generalized to solve Identification to $\mathcal{H}$, along with many other graph modification problems, in the same running time. We only present this more general result in this thesis.

To represent many possible modifications at once, we readapt the problem of $\mathcal{L}$-Replacement to $\mathcal{H}$ from [121] (see Section 1.4). While the authors of [121] define a replacement action $\mathcal{L}$ so that it maps each graph to a collection of graphs of the *same size*, we now generalize the definition so that $\mathcal{L}$ maps each graph to a collection of graphs of *smaller or equal size*. That is, we extend $\mathcal{L}$-Replacement to $\mathcal{H}$ so that it can simulate any modification problem where the modification is a combination of edge removal, edge addition, vertex removal, and vertex identification of bounded size. However, we now require that the replacement action is "hereditary", which essentially means that, when some modification is allowed, then modifying less is also allowed (see Subsection 7.1.1 for a formal definition). With this new definition, $\mathcal{L}$-Replacement to $\mathcal{H}$ with $\mathcal{L}$ hereditary can simulate in particular (non-exhaustively) Vertex Deletion to $\mathcal{H}$, Identification to $\mathcal{H}$, Edge Deletion/Addition/Edition to $\mathcal{H}$, Edge Contraction to $\mathcal{H}$, Independent Set Deletion to $\mathcal{H}$, Matching Deletion/Contraction to $\mathcal{H}$, Connected Vertex Deletion to $\mathcal{H}$, as well as Subgraph Complementation to $\mathcal{H}$.

When $\mathcal{H}$ is a minor-closed graph class, $\mathcal{L}$-Replacement to $\mathcal{H}$ is solvable in time $f(k) \cdot n^2$ by the meta-theorem of [287]. However, as already mentioned, for all these problems other than Vertex Deletion to $\mathcal{H}$, prior to our next result, the only minor-closed graph classes where an *explicit* parametric dependence was known (to our knowledge), if any, were classes of bounded treewidth [165, 210, 218, 219, 307]. We prove the following (see also Theorem 7.1.3).

Theorem 2.3.1. *Let $\mathcal{H}$ be a minor-closed graph class and $\mathcal{L}$ be a hereditary replacement action. Then $\mathcal{L}$-Replacement to $\mathcal{H}$ can be solved in time $2^{k^{\mathcal{O}_{s_\mathcal{H}}(1)}} \cdot n^2$.*

The exponent of k in the running time of Theorem 2.3.1 depends on the maximum size of a graph in $\mathsf{obs}(\mathcal{H})$. Thus, the algorithm of Theorem 2.3.1, while being uniformly FPT in k, is *not* uniform in the target class $\mathcal{H}$, as one needs to know an upper bound on the size of the minor-obstructions. This "meta-non-uniformity" applies to all the algorithms presented in this chapter and the next (Chapter 8), and it is also the case, among many others, of the FPT-algorithm in [284] solving Vertex Deletion to $\mathcal{H}$ in time $2^{k^{\mathcal{O}_{s_\mathcal{H}}(1)}} \cdot n^3$.

In the running time of Theorem 2.3.1, the exponent of k depends very badly on the size of the obstructions of $\mathcal{H}$. However, when we restrict ourself to $\mathcal{H}$ being the graphs embeddable in some surface, we managed to remove the dependence on $\mathcal{H}$ from the exponent (see also Theorem 7.1.4).

Theorem 2.3.2. *Let $\mathcal{G}_\Sigma$ be the class of graphs embeddable in a surface Σ of Euler genus at most g. Then $\mathcal{L}$-Replacement to $\mathcal{G}_\Sigma$ can be solved in time $2^{\mathcal{O}_g(k^9)} \cdot n^2$.*

An ingredient of both algorithm is a dynamic programming algorithm solving the problem parameterized by both k and the treewidth, which may be of independent interest (see also Theorem 7.3.4).

Theorem 2.3.3. *Let $\mathcal{H}$ be a minor-closed graph class. Then $\mathcal{L}$-Replacement to $\mathcal{H}$ can be solved in time $2^{\mathcal{O}(k^2+(k+w)\log(k+w))} \cdot n$ on graphs of treewidth at most w.*

Those results can be found in Chapter 7.

2.4 Elimination distance to minor-closedness

Let us now try to lift our results on modulator of bounded size from modulators of unbounded size (but still bounded bidimensionality). For this, we chose to study Elimination Distance to $\mathcal{H}$ (cf. Section 1.5), where $\mathcal{H}$ is any minor-closed graph class. In other words, the modification is vertex deletion, and the measure on the modulator is the treedepth of its torso.

For Elimination Distance to $\mathcal{H}$ when $\mathcal{H}$ is minor-closed, no explicit parametric dependence is known, with the notable exception of treedepth, for which Reidl, Rossmanith, Villaamil, and Sikdar [256] give an algorithm deciding whether $\mathsf{td}(G) \leq k$ in time $2^{\mathcal{O}(k\cdot\mathsf{tw})} \cdot n$, where $\mathsf{tw} := \mathsf{tw}(G)$ (see also [36]). Using our terminology, and given that $\mathsf{tw}(G) \leq \mathsf{td}(G)$ for every graph G, this yields an FPT-algorithm for Elimination Distance to $\mathcal{G}_\emptyset$, where $\mathcal{G}_\emptyset$ is the class consisting of the empty graph, running in time $2^{\mathcal{O}(k^2)} \cdot n$.

Hence, we give the first algorithm with an explicit parametric dependence for the problem (see also Theorem 8.3.1).

Theorem 2.4.1. *Let $\mathcal{H}$ be a minor-closed graph class. Then there is an algorithm that solves Elimination Distance to $\mathcal{H}$ in time $2^{2^{2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}}}} \cdot n^2$.*

If $\mathcal{H}$ is apex-minor-free, then this algorithm runs in time $2^{2^{\mathcal{O}_{s_{\mathcal{H}}}(k^2\log k)}} \cdot n^2$.

If $\mathcal{H}$ is apex-minor-free, then we give a second algorithm with a better parametric dependence, but a worse dependence on the size of the input graph (see also Theorem 8.4.1).

Theorem 2.4.2. *Let $\mathcal{H}$ be an apex-minor-free graph class. Then there is an algorithm that solves Elimination Distance to $\mathcal{H}$ in time $2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}} \cdot n^3$.*

Again, both these algorithms employ a dynamic programming algorithm that solves the problem on graphs of bounded treewidth, which may be of independent interest (see also Theorem 8.3.2).

Theorem 2.4.3. *Let $\mathcal{H}$ be a minor-closed graph class. Then there is an algorithm that solves Elimination Distance to $\mathcal{H}$ in time $2^{\mathcal{O}(k\cdot w+w\log w)} \cdot n$ on graphs of treewidth at most w.*

The algorithm of Theorem 2.4.3 can be seen as a generalization of the algorithm of Reidl, Rossmanith, Villaamil, and Sikdar [256] deciding whether $\mathsf{td}(G) \leq k$ in time $2^{\mathcal{O}(k\cdot\mathsf{tw})} \cdot n$. Since, for any graph G and any graph class $\mathcal{H}$, $\mathcal{H}\text{-}\mathsf{td}(G) \leq \mathsf{td}(G) \leq \mathsf{tw}(G) \cdot \log n$, Theorem 2.4.3 implies the existence of an XP-algorithm for Elimination Distance to $\mathcal{H}$ parameterized by treewidth, when $\mathcal{H}$ is minor-closed, running in time $n^{\mathcal{O}(\mathsf{tw}^2)}$. Given that the question of whether Treedepth is in FPT parameterized by tw is still open, this is the best type of algorithm that one can expect for Elimination Distance to $\mathcal{H}$ parameterized by treewidth. Furthermore, since $\mathsf{tw} \preceq \mathsf{td}$, Theorem 2.4.3 implies an FPT-algorithm for Elimination Distance to $\mathcal{H}$ parameterized by treedepth, running in time $2^{\mathcal{O}(\mathsf{td}^2)} \cdot n$.

Given that the class $\mathcal{E}_k(\mathcal{H})$ of graphs G such that $\mathcal{H}\text{-}\mathsf{td}(G) \leq k$ is minor-closed when $\mathcal{H}$ is minor-closed, one can once again try to find the (minor-)obstructions of $\mathcal{E}_k(\mathcal{H})$. We prove the following (big but at least explicit) bound on the size of the obstructions (see also Theorem 8.6.1).

Theorem 2.4.4. *Let $\mathcal{H}$ be a minor-closed graph class. Then the obstructions of $\mathcal{E}_k(\mathcal{H})$ have $2^{2^{2^{2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}}}}}$ vertices.*

Moreover, this bound drops to $2^{2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}}}$ when $\mathcal{H}$ is apex-minor-free.

Dvořák, Giannopoulou, and Thilikos prove in [100] that every graph in $\mathsf{obs}(\mathcal{E}_k(\mathcal{G}_\emptyset))$ has at most $2^{2^{k-1}}$ vertices. Hence, our double-exponential bound for apex-minor-free graphs is "as good" as the double-exponential bound for treedepth. Theorem 2.4.4 can be seen as a generalization of the results of Sau, Stamoulis, and Thilikos [285], who provide similar upper bounds for the graphs in $\mathsf{obs}(\mathcal{A}_k(\mathcal{G}))$.

Those results are proved in Chapter 8.

2.5 Odd-minors and bipartite treewidth

In order to study graph classes beyond minor-closed graph classes, a promising option is odd-minor-closed graph classes (cf. Subsection 1.3.2). As mentioned in Section 1.7, a first step towards solving (graph modification) problems on odd-minor-closed graph classes, and later graph modification problems to odd-minor-closedness, is to solve (graph modification) problems on graphs of bounded bipartite treewidth btw.

Let us argue that an algorithm on graphs of bounded bipartite treewidth is more general than an algorithm on graphs of bounded treewidth or that are "almost bipartite". It follows easily from the definition that $\mathsf{btw}(G) = 0$ if and only if G is bipartite (indeed, to prove the sufficiency, just take a single bag containing the whole bipartite graph, with no apex vertices). More generally, for every graph G it holds that $\mathsf{btw}(G) \leq \mathsf{oct}(G)$, where oct denotes the size of a minimum *odd cycle transversal*, that is, a vertex set X such that $G - X$ is bipartite. On the other hand, since a bipartite tree decomposition is a tree decomposition whose width is not larger than the maximum size of a bag (in each bag, just declare all vertices as apices), for every graph G it holds that $\mathsf{btw}(G) \leq \mathsf{tw}(G) + 1$, where tw denotes treewidth. Thus, a graph class of bounded treewidth or bounded oct also has bounded bipartite treewidth.

We designed a general dynamic programming scheme to solve problems on graphs of bounded btw (see Theorem 9.4.1). Using this scheme, we solved several well-known problems and, in particular, several graph modification problems parameterized by btw (given an instance (G, k) where G is a graph on n edges and m vertices). As such, we prove the following (see also Corollary 9.5.6 for Theorem 2.5.1, Corollary 9.5.8 and Corollary 9.5.12 for Theorem 2.5.2, and Corollary 9.5.20 for Theorem 2.5.3). Note that Campbell, Gollin, Hendrey, and Wiederrecht [49] recently announced an FPT-approximation algorithm to construct a bipartite tree decomposition (cf. Proposition 9.3.2).

Theorem 2.5.1. *Let $\mathcal{H}_t$ be the class of graphs that exclude K_t as a subgraph. Then, there is an algorithm that, given a graph G and a bipartite tree decomposition of G of width w, solves* Vertex Deletion to $\mathcal{H}_t$ *in time $\mathcal{O}(2^w \cdot (w^t \cdot (n + m) + m\sqrt{n}))$.*

Theorem 2.5.2. *There is an algorithm that, given a graph G and a bipartite tree decomposition of G of width w, solves* Weighted (resp. Unweighted) Vertex Cover *in time $\mathcal{O}(2^w \cdot (w \cdot (w + m) + m \cdot n))$ (resp. $\mathcal{O}(2^w \cdot (w \cdot (w + m) + m\sqrt{n}))$).*

Theorem 2.5.3. *There is an algorithm that, given a graph G and a bipartite tree decomposition of G of width w, solves* Odd Cycle Transversal *in time $\mathcal{O}(3^w \cdot w \cdot (m + k^2) \cdot n)$.*

Unfortunately, many graph modification problems are unlikely to be solved in FPT-time, or even XP-time, parameterized by bipartite treewidth, as proved in [314]: if H is a bipartite graph containing P_3 as a subgraph and $\mathcal{H}$ is the class of graphs that excludes H as a subgraph, or an induced subgraph, or a minor, or an odd-minor, then Vertex Deletion to $\mathcal{H}$ is NP-complete even on graphs with bipartite treewidth zero, that is, bipartite graphs.

See Chapter 9 for the proof of those results, and see more particularly Table 9.1 for an overview of all other results we actually obtain in [171], concerning problems such as MaxCut and packing problems.

2.6 Global modulators

We explain in Section 1.6 that the current algorithmic meta-theorems for graph modification problems [120, 287] employ the irrelevant vertex technique of [271] that only works up to modulators of bounded bidimensionality. In this part, we abandon efficiency towards generality, and we break the limit of bounded bidimensionality. That is, we consider graph modification problems with a modulator of unbounded bidimensionality, starting with a modulator whose torso is *planar*, and we create a new irrelevant vertex technique that works in this case (cf. Subsection 3.1.3).

Planar modulators

Taking inspiration from the definition of $\mathcal{H}$-p (cf. Section 1.5), we can define $\mathcal{H}$-Planarity as follows (see Figure 2.2 for an illustration).

$\mathcal{H}$-Planarity

Input:	A graph G.
Question:	Is there a set $X \subseteq V(G)$ whose torso is planar and such that $G - X \in \mathcal{H}$?

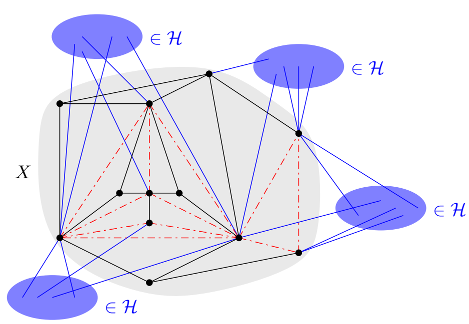

Figure 2.2: A yes-instance of $\mathcal{H}$-Planarity. The additional edge of the torso of X are depicted in dashed red.

The set X may here have unbounded size, or even unbounded bidimensionality, so it escapes the scope of the meta-theorem of [287]. And for this problem, we prove the following.

Theorem 2.6.1. *Let $\mathcal{H}$ be a graph class that is hereditary, CMSO-definable, and decidable in time $\mathcal{O}(n^c)$ for some constant c. Then, there is an algorithm solving $\mathcal{H}$-Planarity in time $\mathcal{O}(n^4 + n^c \log n)$.*

Observe that we do not restrict ourselves here to target classes that are minor-closed, but instead to a far more general family of graph classes.

If we define $\mathsf{p}_{\mathsf{planar}}$ to be the parameter that maps each graph to zero if it is planar, and to its treewidth otherwise, then $\mathcal{H}$-Planarity is the problem of deciding whether $\mathcal{H}\text{-}\mathsf{p}_{\mathsf{planar}} = 0$. As said previously, we know from [6] that, if Vertex Deletion to $\mathcal{H}$ is FPT, then checking whether $\mathcal{H}\text{-}\mathsf{tw}(G) \leq k$ is FPT (where $\mathcal{H}$ has the mild constraints of Theorem 2.6.1 and is closed under disjoint union). Hence, it implies from Theorem 2.6.1 that, if Vertex Deletion to $\mathcal{H}$ is FPT, then the same holds for checking whether $\mathcal{H}\text{-}\mathsf{p}_{\mathsf{planar}}(G) \leq k$ (under the same constraints for $\mathcal{H}$). Therefore, $\mathsf{p}_{\mathsf{planar}}$ is a minor-monotone parameter p such that $\mathsf{hw} \preceq \mathsf{p} \preceq \mathsf{tw}$ and such that the following holds.

> Let $\mathcal{H}$ be a hereditary, CMSO-definable, closed under disjoint union, and polynomial-time decidable graph class such that Vertex Deletion to $\mathcal{H}$ is FPT. Then, given a graph G and $k \in \mathbb{N}$, checking whether $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ is FPT.

As we said in Section 1.5, the above statement is likely to hold for any parameter p such that $\mathsf{tw} \preceq \mathsf{p} \preceq \mathsf{size}$ using the techniques from [6]. So now, the main challenge is focusing on parameters p such that $\mathsf{hw} \preceq \mathsf{p} \preceq \mathsf{tw}$.

After $\mathsf{p}_{\mathsf{planar}}$, we studied two other parameters combining treedepth or treewidth with planarity.

Planar treedepth

We first created a planar variant of treedepth where, instead of removing a single vertex from each connected component at each step, we now remove a set whose torso is planar. The *planar treedepth* of a graph G, denoted by $\mathsf{ptd}(G)$, is defined as follows:

$$\mathsf{ptd}(G) = \begin{cases} 0 & \text{if } G \text{ is the empty graph,} \\ 1 + \min\{\mathsf{ptd}(G - X) \mid X \subseteq V(G), \mathsf{torso}(G, X) \text{ is planar}\} & \text{if } G \text{ is connected,} \\ \max\{\mathsf{ptd}(H) \mid H \text{ is a connected component of } G\} & \text{otherwise.} \end{cases}$$

Then, the $\mathcal{H}$*-planar treedepth* of G is the minimum k such that $\mathcal{H}\text{-}\mathsf{ptd}(G) \leq k$. Said in another way, given that planarity is closed under disjoint union, this corresponds to removing recursively k sets $X_1, \ldots, X_k$ whose torso is planar, such that the remaining components are in $\mathcal{H}$ (see Figure 2.3). Intuitively, $\mathsf{ptd}(G)$ expresses the minimum number of "planar layers" that should be removed in order to obtain the target property $\mathcal{H}$. Using the technique created for Theorem 2.6.1, we prove the

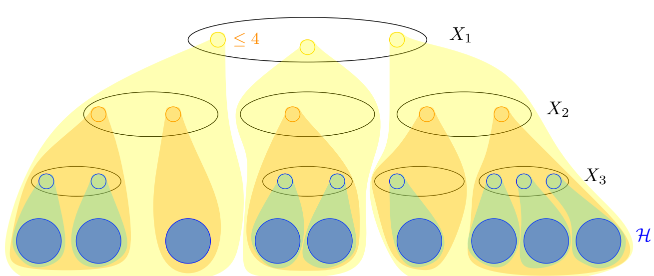

Figure 2.3: A graph of $\mathcal{H}$-planar treedepth at most three.

following.

Theorem 2.6.2. *Let $\mathcal{H}$ be a hereditary and CMSO-definable graph class that is closed under disjoint union. Suppose that there is an* FPT*-algorithm solving* Vertex Deletion to $\mathcal{H}$ *parameterized by* *the solution size k in time $\mathcal{O}_k(n^c)$. Then there is an* FPT*-algorithm that, given a graph G and $k \in \mathbb{N}$, decides whether G has $\mathcal{H}$-planar treedepth at most k in time $\mathcal{O}_k(n^4 + n^c \log n)$.*

Note that, given that we solve problems from uncharted territories, we do not optimize the parametric dependence of our algorithms in this part.

Planar treewidth

We also study a planar variant of treewidth. The *planar width* of a tree decomposition $\mathcal{T}$ is the maximum size (minus one) of a bag of $\mathcal{T}$ whose torso is not planar. The *planar treewidth* of a graph G, denoted by $\mathsf{ptw}(G)$, is the minimum planar width among all tree decompositions of G. This parameter is not new: as mentioned in Section 1.7, given a *singly-crossing* graph H, there is a constant c_H such that, if a graph G excludes H as a minor, then G has planar treewidth at most c_H [268]. The $\mathcal{H}$*-planar treewidth* of G is the minimum k such that $\mathcal{H}\text{-}\mathsf{ptw}(G) \leq k$. Using once again the technique created for Theorem 2.6.1, we prove the following.

Theorem 2.6.3. *Let $\mathcal{H}$ be a hereditary and CMSO-definable graph class that is closed under disjoint union. Suppose that there is an* FPT*-algorithm solving* Vertex Deletion to $\mathcal{H}$ *parameterized by the solution size k in time $\mathcal{O}_k(n^c)$. Then there is an* FPT*-algorithm that, given a graph G and $k \in \mathbb{N}$, decides whether G has $\mathcal{H}$-planar treewidth at most k in time $\mathcal{O}_k(n^4 + n^c \log n)$.*

We refer the reader to Chapter 10 for our results on unbounded bidimensionality modulators as well as applications for instance in designing an algorithm for Counting Perfect Matchings or EPTASes.

The chapters of this thesis are presented ordered by the parameterization on the modulator in Figure 2.4.

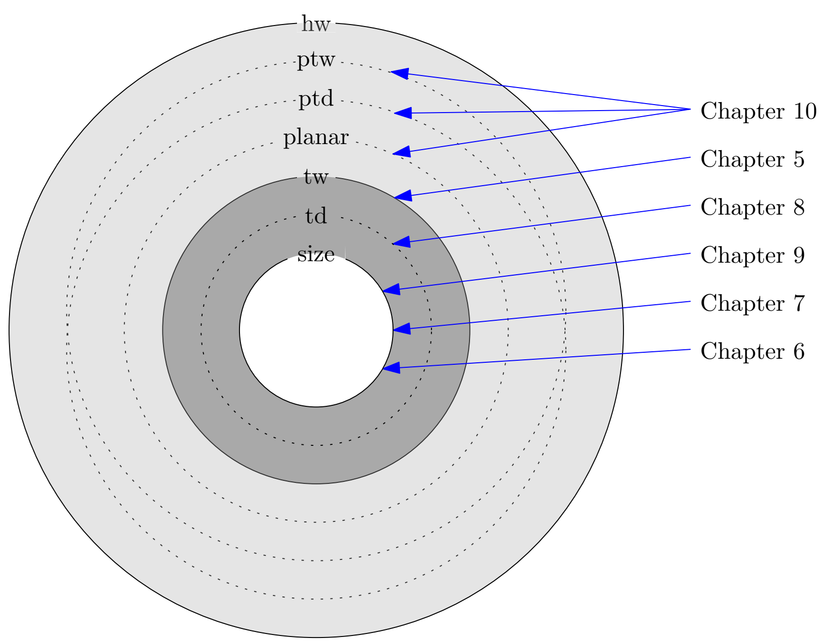

Figure 2.4: Overview of the parameterization of the modulators in the different chapters. Note that ptd and ptw are actually incomparable, and that our result for tw is on its annotated version, while the other results are on the torso-version of the parameters.

2.7 Papers

This thesis covers the following articles (here by chronological order of submission).

In Chapter 8, we present parts of the following:
Laure Morelle, Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos.
Faster parameterized algorithms for modification problems to minor-closed classes,
ICALP 2023: 10.4230/LIPIcs.ICALP.2023.93,
TheoretiCS: 10.48550/arXiv.2210.02167.

In Chapter 9, we present parts the following:
Lars Jaffke, Laure Morelle, Ignasi Sau, and Dimitrios M. Thilikos.
Dynamic Programming on Bipartite Tree Decompositions,
IPEC 2023: 10.4230/LIPIcs.IPEC.2023.26,
JCSS: 10.1016/j.jcss.2025.103722,
arXiv: 10.48550/arXiv.2309.07754.

In Chapter 6, we present the following:
Laure Morelle, Ignasi Sau, and Dimitrios M. Thilikos.
Vertex identification to a forest,
Discrete Mathematics: 10.1016/j.disc.2025.114699,
arXiv: 10.48550/arXiv.2409.08883.

In Chapter 7, we present the following:
Laure Morelle, Ignasi Sau, and Dimitrios M. Thilikos.
Graph modification of bounded size to minor-closed classes as fast as vertex deletion,
ESA 2025: 10.4230/LIPIcs.ESA.2025.7,
arXiv: 10.48550/arXiv.2504.16803, submitted to a journal.

In Chapter 10, we present the following:
Fedor Fomin, Petr Golovach, Laure Morelle, and Dimitrios M. Thilikos.
$\mathcal{H}$-Planarity and Parametric Extensions: when Modulators Act Globally,
arXiv: 10.48550/arXiv.2507.08541, accepted in SODA 2026.

In Chapter 5, we present the following:
Laure Morelle, Evangelos Protopapas, Dimitrios M. Thilikos, and Sebastian Wiederrecht.
Excluding Pinched Spheres,
arXiv: 10.48550/arXiv.2506.14421, submitted to a journal.

The following articles were written during the PhD project, but are not part of the thesis.

Gaétan Berthe, Yoann Coudert-Osmont and Alexander Dobler, Laure Morelle, Amadeus Reinald, and Mathis Rocton.
PACE Solver Description: Touiouidth,
IPEC 2023: 10.4230/LIPIcs.IPEC.2023.38.

Nicolas Bousquet, Quentin Chuet, Victor Falgas-Ravry, Amaury Jacques, and Laure Morelle.

A note on locating sets in twin-free graphs,
Discrete Mathematics: 10.1016/j.disc.2024.114297,
arXiv: 10.48550/arXiv.2405.18162.

Davi de Andrade, Júlio Araújo, Morelle Laure, Ignasi Sau, and Ana Silva.
On the parameterized complexity of computing good edge-labelings,
arXiv: 10.48550/arXiv.2408.15181, submitted to a journal.

Matthias Bentert, Fedor Fomin, Petr Golovach, and Laure Morelle.
When does FTP become FPT?,
arXiv: 10.48550/arXiv.2506.17008, accepted in WG 2025 and submitted to a journal.

Matthias Bentert, Fedor Fomin, Petr Golovach, and Laure Morelle.
Fault-tolerant Matroid Bases,
ESA 2025: 10.4230/LIPIcs.ESA.2025.83,
arXiv: 10.48550/arXiv.2506.22010, submitted to a journal.

Marthe Bonamy, Laure Morelle, Timothé Picavet, and Alexander Scott.
Faster Algorithms for the Pre-Assignment Problem for Unique Minimum Vertex Cover,
submitted to a conference.

CHAPTER 3

Techniques

Contents

To obtain the results of this thesis, we revisit and readapt old techniques, as well as create new ones when necessary. Let us discuss the main techniques employed throughout the thesis.

3.1 Graph modification problems

The key technique to solve problems related to Graph Minor Theory is the *irrelevant vertex technique* developed by Robertson and Seymour [271]. Here is how it usually goes. If our problem Π is such that yes-instances of the problem exclude some bounded size clique as a minor, then this permits us to apply the flat wall theorem (see Subsection 3.1.2) from [194, 271, 286], in order to find a big enough structure called *flat wall* in it (see Subsection 3.1.1). Our algorithm repetitively produces equivalence instances of the problem by removing "irrelevant vertices" in the middle of this wall (see Subsection 3.1.3), until the treewidth is bounded. Then, it suffices to apply Courcelle's theorem [18, 67] or a dynamic programming algorithm (see Subsection 3.1.4) to conclude.

We use a somewhat similar approach in Chapter 7, Chapter 8, and Chapter 10 to solve our graph modification problems, with the addition, sometimes, of an "obligatory set" technique (see Subsection 3.1.5). Let us discuss these techniques more in detail, as well as the generalizations we brought.

3.1.1 Flatness

Let us begin by sketching some necessary notions.

Walls. A *wall* is a subdivided hexagonal grid, such as the one pictured in Figure 3.1. See Subsection 4.6.1 for the formal definition. Notice that a wall can easily be contracted to a grid, and

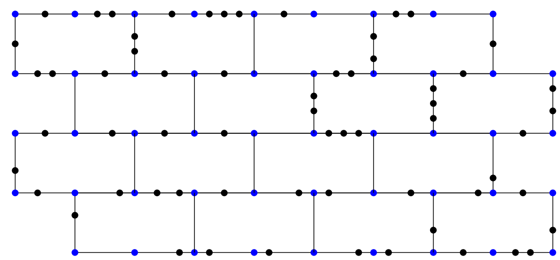

Figure 3.1: A 5-wall. Its first layer is depicted in red and its second layer in orange. Its central vertices are depicted in a green square.

a grid contains a wall as a subgraph. As discussed in Section 1.6 and Subsection 1.7.1, a graph either has bounded treewidth, or contains a big grid as a *minor*. One can actually prove instead that a graph either has bounded treewidth, or contains a wall as a *subgraph*. Subgraphs are easier to work with than minors, so researchers prefer to use walls instead of grids.

Renditions. A *rendition* is essentially a way to draw a graph on a disk such that crossings are localized into cells with at most three vertices on the boundary, as depicted in Figure 3.2. See Section 4.5 for the formal definition. If we find such a drawing, it means that two disjoint paths

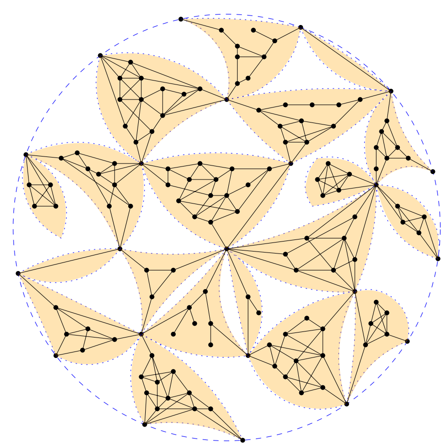

Figure 3.2: A (vortex-free) rendition. The cells are the orange disks with a dotted boundary.

cannot enter a cell, cross one another, and then exit this cell. That is, seen from afar (meaning if we remove the interior of each cell and only add edges between the vertices on the boundary), the graph looks planar, hence the notion of “flatness”.

Flat walls. A *flat wall* is a combination of both a wall and a rendition. More precisely, this is a wall W such that there is a rendition on the disk containing W and whose boundary is the perimeter of W. Also, the perimeter of W separates the “rendition part” from the rest of the graph. See Figure 3.3 for an illustration. This informal definition is missing many crucial elements (see

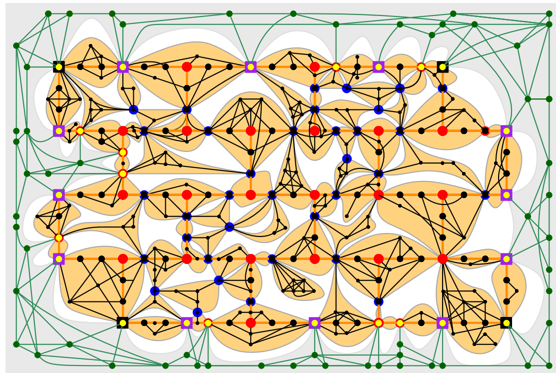

Figure 3.3: A flat wall (picture by Dimitrios M. Thilikos).

Subsection 4.6.2 for the formal definition), but what is crucial to remember it that flat walls retain the nice structural properties of walls (and grids), with the addition of some “flatness” property.

3.1.2 Flat wall theorem

The *flat wall theorem*, in its original form in Robertson and Seymour's Graph Minor series [271], states that a graph G either

- has a big clique as a minor, or
- has bounded treewidth, or
- has a big enough wall W that is flat in $G - A$, where A is a vertex set of small size, called *apex set*.

Since then, several improvements and variants of the flat wall theorem were created [64, 138, 141, 188, 194, 286].

The flat wall theorem is useful for problems whose yes-instances exclude a big clique as a minor, such as $\mathcal{L}$-Replacement to $\mathcal{H}$ or Elimination distance to $\mathcal{H}$ where $\mathcal{H}$ is minor-closed. Hence, in Chapter 7 and Chapter 8, we use the variants from [194, 286] (Proposition 4.6.2 and Proposition 4.6.3) to find a flat wall in the input graph.

In Chapter 10 for $\mathcal{H}$-Planarity however, $\mathcal{H}$ is possibly not minor-closed, and yes-instances of $\mathcal{H}$-Planarity might contain arbitrarily cliques as minors. To solve this issue, we first use the meta-theorem of Lokshtanov, Ramanujan, Saurabh, and Zehavi from [224] and the random-sampling technique from [60], in order to first prove that the problem of solving $\mathcal{H}$-Planarity can be reduced to the problem of solving a restricted version of the problem, called $\mathcal{H}^{(k)}$-Planarity, which may be of independent interest. The task of this problem is to decide, given a graph G and a positive integer k, whether G has an $\mathcal{H}^{(k)}$-planar modulator, where $\mathcal{H}^{(k)}$ is the subclass of $\mathcal{H}$ composed by the graphs of $\mathcal{H}$ with at most k vertices. The yes-instances of $\mathcal{H}^{(k)}$-Planarity are minor-free (and even apex-minor-free) and thus we can apply the flat wall theorem here. More specifically, we require a flat wall of G. To find such a flat wall without apex set A, we need a variant of the flat wall theorem where, instead of searching a clique as a minor, we search for an apex graph as a minor. While it is known that apex-minor-free graphs with big treewidth contain a flat wall [141], we do not know of any algorithmic variant of this result that we may refer to, so we created our own algorithmic variant of the flat wall theorem (Theorem 10.2.2). In Chapter 10, we use both the new variant created for Chapter 10 and the variant from [286].

While we do not use any irrelevant vertex technique to prove the structure theorem of Chapter 5, we do need to find a flat wall, as is often the case for structure theorems [195]. Here, we require again a flat wall of G, without apex set. While we could use the flat wall theorem proved in Chapter 10, given that we exclude an edge-apex graph as a minor, we create a new variant with better constants where we replace the clique by an edge-apex graph (Lemma 5.2.22).

3.1.3 Irrelevant vertex technique

A vertex v such that (G, k) and $(G - v, k)$ are equivalent instances of a problem Π is called an *irrelevant vertex* (for Π). The irrelevant vertex technique was originally developed by Robertson and Seymour in their Graph Minor series [271] to solve the 2-Disjoint Path problem, but was later generalized to many problems related to graph minors, such as those from the following non-exhaustive list [3, 4, 24, 53, 77, 120, 121, 126, 126, 142, 143, 151, 166, 174, 178, 179, 186, 189–191, 229, 284–287, 301]. Its general applicability can be guaranteed by a series of algorithmic meta-theorems that have been recently developed in [120, 146] and, in its more powerful version, in [287]. The irrelevant vertex technique essentially asserts the following: if a graph G contains a small apex set A and a big wall W that is a flat wall of $G - A$, then one can find a vertex v of W that is irrelevant. Hence, we can remove this vertex v and recurse.

Bounded bidimensionality modifications. Typically, the applicability of the irrelevant vertex technique requires that the modifications are affecting only a *restricted area* of a big flat wall, so that we find a smaller flat wall avoiding the modified vertices. In technical terms, according to [287], a sufficient condition is that the modifications affect a set of vertices with low bidimensionality, such as is the case in the algorithmic meta-theorems in [120, 146, 287].

In particular, the authors of [284, 285] adapt the original irrelevant vertex technique to Vertex Deletion to $\mathcal{H}$ when $\mathcal{H}$ is minor-closed, and more generally, to any problem where one decides whether $\mathcal{H}\text{-}\mathsf{p} \leq k$ for $\mathsf{tw} \preceq \mathsf{p} \leq \mathsf{size}$ (see Proposition 8.2.4). Given that $\mathsf{tw} \preceq \mathsf{td}$ and that the running time of the algorithm of [284, 285] comes with a nice parametric dependence, we use their result (Proposition 8.2.4) for Elimination Distance to $\mathcal{H}$ in Chapter 8 to find an irrelevant vertex in a flat wall.

For $\mathcal{L}$-Replacement to $\mathcal{H}$ in Chapter 7, we cannot use the result of [284, 285] as a blackbox, as our modifications are more general than just deleting vertices. Hence, we prove a more general irrelevant vertex result that works for $\mathcal{L}$-Replacement to $\mathcal{H}$ when $\mathcal{H}$ is minor-closed (Theorem 7.3.1) and another one with a better parametric dependence when $\mathcal{H}$ has bounded genus (Theorem 7.3.2).

Global modifications. In Chapter 10, the modifications are *global* and they may occur everywhere along the two-dimensional area of the flat wall. Therefore, precisely because our modulator has potentially unbounded bidimensionality, it is not possible to apply any of the results in [120, 146, 287] in order to find an irrelevant vertex. This initiates the challenge of applying the irrelevant vertex technique under the setting that the modification potentially affects any part of the flat wall. Hence, we need to, and we did, create a new irrelevant vertex technique beyond the meta-algorithmic framework of [287] which, we believe, may have independent use for other problems involving global modification (Corollary 10.2.22). The rough idea is that, given a vertex v in the center of the flat wall W, if we find a solution restricted to W, and another one in $G - v$, then we may glue these two solutions on a disk inside W to obtain a solution for G. The main technicality is how to glue solutions correctly, which requires topological arguments.

3.1.4 Bounded treewidth

When the input graph has bounded treewidth, the meta-theorem of Courcelle [18, 67] (see also Section 1.6 and Proposition 4.3.2) is often enough to conclude. This is for instance what we do in Chapter 10. However, Courcelle's theorem does not give an explicit dependence on $|\varphi| + \mathsf{tw}$. To get this explicit parametric dependence, we have to design instead dynamic programming algorithms (DP) on tree decompositions.

Representative-based technique. Baste, Sau, and Thilikos [23] developed a representative-based technique that gives a DP solving VERTEX DELETION TO $\mathcal{H}$ when $\mathcal{H}$ is minor-closed in efficient FPT-time parameterized by treewidth (cf. Section 4.4, Proposition 4.4.1 and Proposition 4.4.2). We use the technique of [23] to develop in Chapter 7 (Section 7.6) a DP solving this time the more general problem of $\mathcal{L}$-REPLACEMENT TO $\mathcal{H}$ (Theorem 2.3.3). Additionally, we combine in Chapter 8 (Section 8.5) the representative-based technique of [23] with the DP of [256] deciding whether the treedepth is at most k in FPT-time parameterized by $\mathsf{tw} + k$ (Proposition 8.5.4), to create the first DP deciding whether the elimination distance to $\mathcal{H}$ is at most k in in FPT-time parameterized by $\mathsf{tw} + k$ (Theorem 2.4.3).

DP for other treewidth-related parameters. In Chapter 9, we go further and develop DPs on tree decompositions where the parameter is not the treewidth anymore, but the bipartite treewidth instead. Given that bipartite treewidth is a relatively unknown parameter, there is no Courcelle-like meta-theorem expliciting the types of problems that may be solved efficiently parameterized by btw. Hence, we created our own self-standing general DP scheme to solve problems efficiently parameterized by btw (Theorem 9.4.1), and applied it to various problems.

3.1.5 Obligatory sets

In Chapter 7, if we just apply the flat wall theorem to find an irrelevant vertex and recurse until the treewidth is bounded, we will not obtain the parametric dependence claimed in Theorem 2.3.1. To achieve this running time, we use an additional technique. If we find in our graph a "sunflower" of $k+1$ obstructions of the target class $\mathcal{H}$ that intersect in some small vertex set A, called *obligatory set*, then it implies that some modification must be done in A, since otherwise, one of the $k+1$ obstructions will remain in the modified graph (see Figure 3.4 and Lemma 7.3.3). Hence, we may

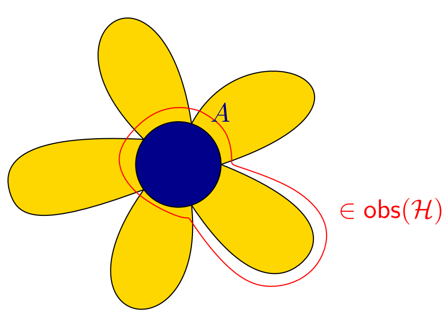

Figure 3.4: A "sunflower" composed of obstructions of $\mathcal{H}$ intersecting in a vertex set A.

simply use branching to guess which part of A is modified and how, and then apply this partial modification and recurse. Given that the modification has bounded size, the depth of the search tree will not be too big. This sunflower technique is inspired from [229] and in our case, is a generalization from [285]. The main technicality in our case is to find such a sunflower efficiently in the input graph.

In Chapter 8 unfortunately, we cannot use this technique, given that the modification has unbounded size, and thus that the depth of the search tree might be unbounded. This explains the triple exponential dependence on k of Theorem 2.4.1. The only case when we may still use the sunflower technique is when the target class is apex-minor-free. In this case, the obligatory set A that we find is a singleton. Therefore, there is no need to branch, and the only possible modification, if any, is to delete A. Hence, the improved running time of Theorem 2.4.2.

3.2 Structure theorem

The graph minor structure theorem of [275] (cf. Subsection 1.7.1), and more specifically the self-contained proof of [194, 195], gives a global strategy to design a structure theorem. Using the flat wall theorem, we find a flat wall W inside the input graph G. Inside W, as depicted in Figure 3.3, there is a drawing of the graph such that crossings are localized, in the sense that they only happen in *cells* with at most three vertices on the boundary. We put the rest of the graph (whatever is outside of W) inside a single cell whose boundary contains the vertices in the perimeter of W. Such a cell with at least four vertices is called a *vortex*. The goal is then, using techniques from [195], to split the vortex into smaller vortices, until we obtain a flat drawing in some surface (in our case the projective plane; cf. Theorem 2.1.1) with only a few vortices of small "depth", where the depth essentially corresponds to the number of crossings that may "escape" from the vortex. Such an almost flat drawing in a surface Σ is called a Σ-*decomposition* of *breadth* b (the number of vortices) and *depth* d (the maximum depth of a vortex). Hence, we obtain what we call a *local structure theorem*, that is a statement along the lines of:

> There exist a surface Σ_H and constants b_H and d_H such that, if G is H-minor-free, then there is a Σ_H-decomposition of G of breadth at most b_H and depth at most d_H.

From this, the goal is to obtain a *(global) structure theorem* stated along the lines of:

> There exist a surface Σ_H and constants b_H and w_H such that, if G is H-minor-free, then there is a tree decomposition of G such that the torso of each bag has an "almost embedding" in Σ_H of breadth at most b_H and "width" at most w_H.

An *almost embedding* of G in Σ is similar to a Σ-decomposition, but where the non-vortex cells have no vertex drawn in their interior. To obtain an almost embedding from a Σ-decomposition, we essentially push out the interior of cells to create new bags of the tree decomposition (hence the tree decomposition). Also, the parameter we want to bound on vortices is not their depth anymore, but their *width*, which is essentially their pathwidth. To bound the width of vortices instead of their depth, we once again push out (more carefully) vertices in the interior of vortices to new bags.

The described strategy is essentially enough to prove that the graph induced by each bag $\beta(t)$ has an almost embedding of small depth and breadth. The problem in our case is to prove the same for the *torso* of $\beta(t)$. In most structure theorems, such as the ones of [101, 195], what is actually stated is that the torso of each bag has an almost embedding in Σ_H of breadth at most b_H and "width" at most w_H *after the removal of* a_H *vertices*. While we do not enter the details here, those a_H vertices, called *apices*, make it easy to deduce an almost embedding of $\mathsf{torso}(G, \beta(t))$ from an almost embedding of $G[\beta(t)]$, by simply adding a bounded number of apices. In our case, this method does not work, given that we want no apices. Hence, we created a new "local-to-global" technique to go from our local theorem to the global one. The idea, without entering into detail, is to create new vortices whose depth is still bounded to handle the "torsification" (see Subsection 5.2.3, Theorem 5.2.32).

From this global theorem, it is easy to obtain Theorem 2.1.2 by identifying each vortex to a single point.

As for Theorem 2.1.3, we find, for any $k \in \mathbb{N}$, a graph embeddable in the pinched sphere that cannot be made projective by identifying a vertex set of bidimensionality at most k. Then, using techniques of [302], we derive Theorem 2.1.3.

CHAPTER 4

Preliminaries on graphs

In this chapter, we formally define the notions that will be used throughout this work and we give some preliminary results.

Contents

4.1 Sets and functions

Let us first give some basic definitions and notations on sets and functions.

Sets and integers. We denote by $\mathbb{N}$ the set of non-negative integers. Given two integers p, q, where $p \leq q$, we denote by $[p, q]$ the set $\{p, \ldots, q\}$. For an integer $p \geq 1$, we set $[p] = [1, p]$ and $\mathbb{N}_{\geq p} = \mathbb{N} \setminus [0, p-1]$. For a set S, we denote by 2^S the set of all subsets of S and, given an integer $r \in [|S|]$, we denote by $\binom{S}{r}$ the set of all subsets of S of size r and by $\binom{S}{\leq r}$ (resp. $\binom{S}{<r}$) the set of all subsets of S of size at most r (resp. $r-1$). If $\mathcal{S}$ is a collection of objects where the operation $\cup$ is defined, then we denote $\bigcup \mathcal{S} = \bigcup_{X \in \mathcal{S}} X$. The function $\mathsf{odd} : \mathbb{R} \to \mathbb{N}$ maps x to the smallest odd non-negative integer larger than x.

Functions. Given a set A, we denote the identity function mapping each $a \in A$ to itself by id_A. Given two sets A, B, $v \in B$, $S \subseteq B$, and a function $f : A \to B$, $f^{-1}(v)$ is the set of elements $u \in A$ such that $f(u) = v$, and $f^{-1}(S) = \bigcup_{v \in S}\{f^{-1}(v)\}$. If B is a collection of objects where the operation $+$ is defined, then, given $R \subseteq A$, we set $f(R) = \sum_{r \in R} f(r)$. Otherwise, given $R \subseteq A$, we set $f(R) = \bigcup_{r \in R}\{f(r)\}$, and the restriction of f to R is denoted by $f|_R$. Additionally, for some new vertex $u \notin A$, $f \cup (u \mapsto v) : A \cup \{u\} \to B$ is the function that maps u to v and whose restriction to A is f. Given two sets A and B, and two functions $f, g : A \to 2^B$, we denote by $f \cup g$ the function that maps $x \in A$ to $f(x) \cup g(x) \in 2^B$. Let $f : A \to B$ be an injective function. Let $K \subseteq B$ be the image of f. By convention, if f is referred to as a bijection, it means that f maps A to K.

Partitions. Given $p \in \mathbb{N}$, a *p-partition* of a set X is a tuple $(X_1, \ldots, X_p)$ of pairwise disjoint subsets of X such that $X = \bigcup_{i \in [p]} X_i$. We denote by $\mathcal{P}_p(X)$ the set of all p-partitions of X. Given a partition $\mathcal{X} \in P_p(X)$, its domain X is also denoted as $\cup\mathcal{X}$. A *partition* is a p-partition for some $p \in \mathbb{N}$. Given $Y \subseteq X$, $\mathcal{X} = (X_1, \ldots, X_p) \in \mathcal{P}_p(X)$, and $\mathcal{Y} = (Y_1, \ldots, Y_p) \in \mathcal{P}_p(Y)$, we say that $\mathcal{Y} \subseteq \mathcal{X}$ if $Y_i \subseteq X_i$ for each $i \in [p]$. Given a set U, two subsets $X, A \subseteq U$, and $\mathcal{X} = (X_1, \ldots, X_p) \in \mathcal{P}_p(X)$, $\mathcal{X} \cap A$ denotes the partition $(X_1 \cap A, \ldots, X_p \cap A)$ of $X \cap A$. Given two disjoint sets X and Y, and $\mathcal{X} = (X_1, \ldots, X_p) \in \mathcal{P}(X)$ and $\mathcal{Y} = (Y_1, \ldots, Y_q) \in \mathcal{P}(Y)$, $\mathcal{X} \cup \mathcal{Y}$ denotes the partition $(X_1, \ldots, X_p, Y_1, \ldots, Y_q) \in \mathcal{P}(X \cup Y)$.

4.2 Basic concepts on graphs

Let us now give some basic definitions on graphs. We use standard graph-theoretic notation and we refer the reader to [87] for any undefined terminology on graphs.

Basic notations on graphs. A graph G is a pair (V, E) where V is a finite set and $E \subseteq \binom{V}{2}$, i.e., all graphs in this thesis are undirected, finite, and without loops or parallel edges. For convenience, we use uv (or vu) instead of $\{u, v\}$ to denote an edge of a graph. We write $\mathcal{G}_{\mathsf{all}}$ for the set of all graphs. We also define $V(G) = V$ and $E(G) = E$. We always use $n = |G|$ for the size of G, i.e., the cardinality of $V(G)$, and m for the cardinality of $E(G)$, where G is the input graph of the problem under consideration.

For $S \subseteq V(G)$, we set $G[S] = (S, E \cap \binom{S}{2})$ and use the shortcut $G - S$ to denote $G[V(G) \setminus S]$. We may also use $G - v$ instead of $G - \{v\}$ for $v \in V(G)$. We say that $G[S]$ is an *induced (by S) subgraph* of G. The *detail* of G is $\max\{|V(G)|, |E(G)|\}$. Given $A, B \subseteq V(G)$, we also denote by $E_G(A, B)$ the set of edges of G with one endpoint in A and the other in B. We may also use $E(A, B)$ instead of $E_G(A, B)$ when there is no risk of confusion. We say that $E' \subseteq E(G)$ is an *edge cut* of G if there is a partition (A, B) of $V(G)$ such that $E' = E(A, B)$. Given two graphs G_1 and G_2, we denote $G_1 \cup G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. Given a graph G, we define the *set of partitions* of G to be the set $\mathcal{P}(G) := \{\mathcal{X} \in \mathcal{P}(X) \mid X \subseteq V(G)\}$.

Some graphs. For $t \in \mathbb{N}$, the *complete graph* K_t (alternatively a *clique* of size t) is the graph with t vertices and all edges between vertices. For $a, b \in \mathbb{N}$, the *complete bipartite graph* $K_{a,b}$ is the graph whose vertex set is the union of a vertex set A of size a and a vertex set B of size b and whose edge set is $E(A, B)$. A *bipartite graph* is a graph that is a subgraph of a complete bipartite graph. For $k \in \mathbb{N}$, the *path* P_k is the graph with k vertices $v_1, \ldots, v_k$ and edges $v_i vi + 1$ for $i \in [k-1]$. We say that P_k has length $k - 1$ (i.e., the *length* of a path is its number of edges). For $k \in \mathbb{N}$, the *cycle* C_k is the graph obtained from the path P_k by adding an edge $v_1 v_k$. A *planar graph* is a graph that can be drawn on a plane without edge crossing.

Neighbors. Given a vertex $v \in V(G)$, we denote by $N_G(v)$ the set of vertices of G that are adjacent to v in G and we set $N_G[v] = N_G(v) \cup \{v\}$. Also, given a set $S \subseteq V(G)$, we set $N_G[S] = \bigcup_{v \in S} N_G[v]$ and $N_G(S) = N_G[S] \setminus S$. A vertex $v \in V(G)$ is *isolated* if $N_G(v) = \emptyset$. The *degree* of a vertex $v \in V(G)$ is its number $|N_G(v)|$ of neighbors.

Connectivity. A graph G is *connected* if every pair of vertices of G is joined by a path. A *connected component* of a graph G is a connected subgraph of G of maximum size. We denote by $\mathsf{cc}(G)$ the connected components of G. A *bridge* (resp. *cut vertex*) in G is an edge (resp. a vertex) whose removal increases the number of connected components of G. Given $k \in \mathbb{N}_{\geq 1}$, we say that a graph G is *k-connected* if, for any set X of size at most $k-1$, $G - X$ is connected. A *block* of G is a maximal connected subgraph of G without a cut vertex.

Separations. A *separation* of a graph G is a pair $(L, R) \in 2^{V(G)} \times 2^{V(G)}$ such that $L \cup R = V(G)$ and there is no edge in G between $L \setminus R$ and $R \setminus L$. The *order* of (L, R) is $|L \cap R|$.

Some graph properties. We say that a graph class $\mathcal{H}$ is *hereditary* (resp. *monotone*) if it contains all the induced subgraphs (resp. subgraphs) of its graphs. A class $\mathcal{H}$ is *closed under disjoint union* if it contains the disjoint union of every two of its graphs.

Torsos and clique-sums. The *torso* of a set $X \subseteq V(G)$, denoted by $\mathsf{torso}(G, X)$, is the graph obtained from $G[X]$ by making $N_G(V(C))$ a clique for each $C \in \mathsf{cc}(G - X)$. Given two graphs G_1 and G_2, and $q \in \mathbb{N}$, a *q-clique-sum* of G_1 and G_2 is obtained from their disjoint union by identifying a q-clique of G_1 with a q-clique of G_2, and then possibly deleting some edges of that clique. A graph class $\mathcal{G}$ is *closed under q-clique-sums* if for each $G_1, G_2 \in \mathcal{G}$, any q-clique-sum of G_1 and G_2 also belongs to $\mathcal{G}$.

Dissolutions and subdivisions. Given a vertex $v \in V(G)$ of degree two with neighbors u and w, we define the *dissolution* of v to be the operation of deleting v and, if u and w are not adjacent, adding the edge uw. Given two graphs H and G, we say that H is a *dissolution* of G if H can be obtained from G after dissolving vertices of G. Given an edge $e = uv \in E(G)$, we define the *subdivision* of e to be the operation of deleting e, adding a new vertex w and making it adjacent to u and v. Given two graphs H and G, we say that H is a *subdivision* of G if H can be obtained from G by subdividing edges. Observe that G is a subdivision of H iff H is a dissolution of G.

Colorings. A *coloring* of a graph G is a function $c : V(G) \to \mathbb{N}$. Given $v \in V(G)$, $c(v)$ is called the *color* of v by c. Given $k \in \mathbb{N}$, a *k-coloring* is a coloring $c : V(G) \to [k]$. Given a coloring c of a graph G and an edge $uv \in E(G)$, we say that uv is *monochromatic* if $c(u) = c(v)$. Otherwise, we say the uv is *bichromatic*. A coloring c of a graph G is said to be *proper* if every edge of G is bichromatic. We say that a graph G is *k-colorable* if there exists a proper k-coloring of G.

Contractions and minors. The *contraction* of an edge $e = uv \in E(G)$ results in a graph G/e obtained from $G \setminus \{u, v\}$ by adding a new vertex w adjacent to all vertices in the set $N_G(\{u, v\})$. Vertex w is called the *heir* of e. A graph H is a *minor* of a graph G, denoted by $H \preceq_{\mathsf{m}} G$, if H can be obtained from a subgraph of G after a series of edge contractions. Equivalently, H is a minor of G if there is a collection $\mathcal{S} = \{S_v \mid v \in V(H)\}$ of pairwise-disjoint connected subsets of $V(G)$, called *branch sets* such that, for each edge $xy \in E(H)$, the set $S_x \cup S_y$ is connected in $V(G)$. The collection $\mathcal{S}$ is called a *model* of H in G and, for each $x \in V(H)$, the set S_x is called *model of x* in

G. Given a finite collection of graphs $\mathcal{F}$ and a graph G, we use notation $\mathcal{F} \preceq_{\mathsf{m}} G$ to denote that some graph in $\mathcal{F}$ is a minor of G.

Minor-closed graph classes. A graph class $\mathcal{H}$ is *minor-closed* if it contains all the minors of its graphs. Given a collection of graphs $\mathcal{F}$, we denote by $\mathsf{exc}(\mathcal{F})$ the class of graphs that do not contain a graph in $\mathcal{F}$ as a minor. Obviously, $\mathsf{exc}(\mathcal{F})$ is minor-closed. We call each graph in $\mathsf{exc}(\mathcal{F})$ *$\mathcal{F}$-minor-free*. A *(minor-)obstruction* of a graph class $\mathcal{H}$ is a graph F that is not in $\mathcal{H}$, but whose minors are all in $\mathcal{H}$. The set of all the obstructions of $\mathcal{H}$ is denoted by $\mathsf{obs}(\mathcal{H})$. By the seminal work of Robertson and Seymour [278], if $\mathcal{H}$ is a minor-closed graph class, then $\mathsf{obs}(\mathcal{H})$ is finite. Note that, if $\mathcal{F} = \mathsf{obs}(\mathcal{H})$, then $\mathsf{exc}(\mathcal{F}) = \mathcal{H}$. We use $\mathcal{G}_\emptyset$ for the graph class containing only the empty graph $G_\emptyset$. Notice that $\mathsf{obs}(\mathcal{G}_\emptyset) = \{K_1\}$.

Thomason proved in [305] that graphs excluding some graph as a minor have the following density property.

Proposition 4.2.1 ([305]). *Let $t \in \mathbb{N}$. If a graph is K_t-minor-free, then it has $\mathcal{O}(t\sqrt{\log t} \cdot n)$ edges.*

Apex number. The *apex number* of a graph G is the smallest integer a for which there is a set $A \subseteq V(G)$ of size a such that $G - A$ is planar.

Identifications. Let G be a graph and $X \subseteq V(G)$. The *identification of* X in G, denoted by $G /\!\!/ X$, is the result of the operation that transforms G into a graph G' obtained from G by deleting X and adding instead a new vertex x adjacent to every vertex in $N_G(X)$. The vertex x is called the *heir* of X. Note that, if $X = \{u, v\}$ with $uv \in E(G)$, then this corresponds to the contraction of uv.

Let $\mathcal{X} = (X_1, \ldots, X_p) \in \mathcal{P}(G)$. The *identification of* $\mathcal{X}$ in G is the graph $G /\!\!/ \mathcal{X} := G /\!\!/ X_1 /\!\!/ X_2 /\!\!/ \ldots /\!\!/ X_p$. Note that the ordering of the members of the partition does not matter in this definition.

We denote by $\mathcal{I}(G, S)$ the set of all graphs G' that can be obtained by a sequence of identifications of vertices of S. In other words, for each $G' \in \mathcal{I}(G, S)$, there is a partition $\mathcal{P} = (S_1, \ldots, S_r) \in \mathcal{P}(X)$ such that $G' = G /\!\!/ \mathcal{P}$.

4.3 Tree decompositions

Let us give some more definitions and results related to treewidth.

Tree decompositions. A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \beta)$ where T is a tree and $\beta \colon V(T) \to 2^{V(G)}$ is a function, whose images are called the *bags* of $\mathcal{T}$, such that

- $\bigcup_{t \in V(T)} \beta(t) = V(G)$,
- for every $e \in E(G)$, there exists $t \in V(T)$ with $e \subseteq \beta(t)$, and
- for every $v \in V(G)$, the set $\{t \in V(T) \mid v \in \beta(t)\}$ induces a subtree of T.

The *width* of $\mathcal{T}$ is the maximum size of a bag minus one and the *treewidth* of G, denoted by $\mathsf{tw}(G)$, is the minimum width of a tree decomposition of G. Given $tt' \in E(T)$, the *adhesion* of t and t', denoted by $\mathsf{adh}(t, t')$, is the set $\beta(t) \cap \beta(t')$. The *adhesion* of a node $t \in V(T)$ is the maximum adhesion of t and t' over all neighbors t' of t and the *adhesion* of $\mathcal{T}$ is the maximum adhesion of a node of $\mathcal{T}$. The *torso* of $\mathcal{T}$ at node t is the graph obtained from $G[\beta(t)]$ by making a clique out of each adhesion of t with a neighbor t'.

A *rooted tree decomposition* is a triple (T, β, r) where (T, β) is a tree decomposition and (T, r) is a rooted tree.

To compute a tree decomposition of a graph of bounded treewidth, we can use the single-exponential 2-approximation algorithm for treewidth of Korhonen [203].

Proposition 4.3.1 ([203]). *There is an algorithm that, given a graph G and an integer k, outputs either a report that $\mathsf{tw}(G) > k$, or a tree decomposition of G of width at most $2k+1$ with $\mathcal{O}(n)$ nodes. Moreover, this algorithm runs in time $2^{\mathcal{O}(k)} \cdot n$.*

As mentioned in Section 1.6, Courcelle's theorem implies that many problems are solvable in linear time on graphs of bounded treewidth.

Proposition 4.3.2 (Courcelle's Theorem [18, 67]). *Let $\mathcal{H}$ be a graph class that is definable by a CMSO formula φ. There is a function $f : \mathbb{N}^2 \to \mathbb{N}$ and an algorithm that, given a graph G of treewidth at most tw, checks whether $G \in \mathcal{H}$ in time $f(|\varphi|, \mathsf{tw}) \cdot n$.*

To describe our dynamic programming algorithms presented in Section 7.6 and Section 8.5, we need a particular type of tree decompositions, namely nice tree decompositions.

Nice tree decompositions. A *nice tree decomposition* of a graph G is a rooted tree decomposition (T, β, r) such that:

- every node has either zero, one or two children,
- if x is a leaf of T, then $\beta(x)$ is a singleton (x is a *leaf node*),
- if x is a node of T with a single child y, then $|\beta(x) \setminus \beta(y)| = 1$ (x is an *introduce node*) or $|\beta(y) \setminus \beta(x)| = 1$ (x is a *forget node*), and
- if x is a node with two children x_1 and x_2, then $\beta(x) = \beta(x_1) = \beta(x_2)$ (x is a *join node*).

To find a nice tree decomposition from a given tree decomposition, we use the following well-known result proved, for instance, in [12].

Proposition 4.3.3 ([12]). *Given a graph G with n vertices and a tree decomposition (T, β) of G of width w, there is an algorithm that computes a nice tree decomposition of G of width w with at most $\mathcal{O}(w \cdot n)$ nodes in time $\mathcal{O}(w^2 \cdot (n + |V(T)|))$.*

The following result has been proved by Adler, Dorn, Fomin, Sau, and Thilikos in [2].

Proposition 4.3.4 ([2]). *There is an algorithm that, given a graph G on m edges, a graph H on h edges without isolated vertices, and a tree decomposition of G of width at most k, outputs, if it exists, a minor of G isomorphic to H. Moreover, this algorithm runs in time $2^{\mathcal{O}(k \log k)} \cdot h^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(h)} \cdot m$.*

4.4 Boundaried graphs

In this part, we define boundaried graphs and notions related to them.

Boundaried graphs are in particular needed for the representative-based technique of [23] that is mentioned in Subsection 3.1.4 and that is used in Section 7.6 and Section 8.5. The idea is the following. Given a minor-closed graph class $\mathcal{H}$, for any graph G with a small boundary $B \subseteq V(G)$, there exist a "representative" graph $\mathsf{Rep}(G)$ of small size (depending on $\mathcal{H}$ and $|B|$) and with same boundary B, such that, for any graph H with boundary B, the graph $G \oplus H$ obtained by gluing G

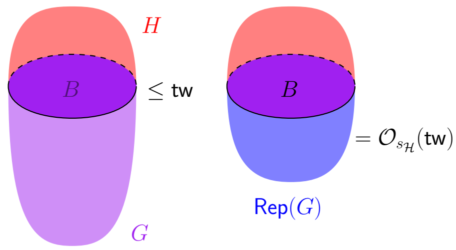

Figure 4.1: Illustration of the representative-based technique of [23].

and H on B is in $\mathcal{H}$ if and only if $\mathsf{Rep}(G) \oplus H$ is in $\mathcal{H}$ (see Figure 4.1). This technique is used in dynamic programming algorithms on tree decompositions in order to be able to keep, at each node t of the tree decomposition, information about a small graph instead of the entire graph induced by the tree rooted at t.

Boundaried graphs. Let $t \in \mathbb{N}$. A *t-boundaried graph* is a triple $\mathbf{G} = (G, B, \rho)$ where G is a graph, $B \subseteq V(G)$, $|B| = t$, and $\rho : B \to \mathbb{N}$ is an injective function. We say that B is the *boundary* of $\mathbf{G}$ and we write $B = \mathsf{bd}(G)$. In other words, a t-boundaried graph is a graph G with a boundary B of t vertices and a labeling ρ of the vertices of B. We call $\mathbf{G}$ *trivial* if all its vertices belong to the boundary. We say that two t-boundaried graphs $\mathbf{G}_1 = (G_1, B_1, \rho_1)$ and $\mathbf{G}_2 = (G_2, B_2, \rho_2)$ are *isomorphic* if $\rho_1(B_1) = \rho_2(B_2)$ and there is an isomorphism from G_1 to G_2 that extends the bijection $\rho_2^{-1} \circ \rho_1$. The triple (G, B, ρ) is a *boundaried graph* if it is a t-boundaried graph for some $t \in \mathbb{N}$. We denote by $\mathcal{B}^t$ the set of all (pairwise non-isomorphic) t-boundaried graphs.

Equivalent boundaried graphs and representatives. We say that two boundaried graphs $\mathbf{G}_1 = (G_1, B_1, \rho_1)$ and $\mathbf{G}_2 = (G_2, B_2, \rho_2)$ are *compatible* if $\rho_2^{-1} \circ \rho_1$ is an isomorphism from $G_1[B_1]$ to $G_2[B_2]$. Given two compatible boundaried graphs $\mathbf{G}_1 = (G_1, B_1, \rho_1)$ and $\mathbf{G}_2 = (G_2, B_2, \rho_2)$, we define $\mathbf{G}_1 \oplus \mathbf{G}_2$ as the graph obtained if we take the disjoint union of G_1 and G_2 and, for every $i \in [|B_1|]$, we identify vertices $\rho_1^{-1}(i)$ and $\rho_2^{-1}(i)$. We also define $\mathbf{G}_1 \bigoplus \mathbf{G}_2$ as the *boundaried* graph $(\mathbf{G}_1 \oplus \mathbf{G}_2, B_1, \rho_1)$. Given $h \in \mathbb{N}$, we say that two boundaried graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ are *h-equivalent*, denoted by $\mathbf{G}_1 \equiv_h \mathbf{G}_2$, if they are compatible and, for every graph H with detail at most h and every boundaried graph $\mathbf{F}$ compatible with $\mathbf{G}_1$ (hence, with $\mathbf{G}_2$ as well), it holds that

$$H \preceq_{\mathsf{m}} \mathbf{F} \oplus \mathbf{G}_1 \iff H \preceq_{\mathsf{m}} \mathbf{F} \oplus \mathbf{G}_2.$$

Note that $\equiv_h$ is an equivalence relation on $\mathcal{B}$. A minimum-sized (in terms of number of vertices) element of an equivalent class of $\equiv_h$ is called *representative* of $\equiv_h$. For $t \in \mathbb{N}$, a *set of t-representatives* for $\equiv_h$, denoted by $\mathcal{R}_h^t$, is a collection containing a minimum-sized representative for each equivalence class of $\equiv_h$ restricted to $\mathcal{B}^t$.

The following results were proved by Baste, Sau, and Thilikos [24] and give a bound on the size of a representative and on the number of representatives for this equivalence relation, respectively.

Proposition 4.4.1 ([24]). *For every $t \in \mathbb{N}$, $q, h \in \mathbb{N}_{\geq 1}$, and $\boldsymbol{G} = (G, B, \rho) \in \mathcal{R}_h^t$, if G is does not contain K_q as a minor, then $|V(G)| = \mathcal{O}_{q,h}(t)$.*

Proposition 4.4.2 ([24]). *For every $t \in \mathbb{N}_{\geq 1}$, $|\mathcal{R}_h^t| = 2^{\mathcal{O}_h(t \log t)}$.*

Moreover, given a boundaried graph of bounded size, the following lemma gives an algorithm to find its representative. While this is might be considered folklore, we include here its proof for the sake of completeness.

Lemma 4.4.3. *Given a finite collection of graphs $\mathcal{F}$, $h, t, k \in \mathbb{N}$, the set $\mathcal{R}$ of representatives in $\mathcal{R}_h^t$ whose underlying graphs are $\mathcal{F}$-minor-free, and a t-boundaried graph $\mathbf{G}$ with k vertices whose underlying graph is $\mathcal{F}$-minor-free, there is an algorithm that outputs the representative of $\mathbf{G}$ in $\mathcal{R}$ in time $2^{\mathcal{O}_{\ell_{\mathcal{F}},h}(t\log t+\log(k+t))}$.*

Proof. Let $\mathcal{H}$ be the set of graphs with detail at most h. For a t-boundaried graph $\mathbf{G}$ of size k whose underlying graph is $\mathcal{F}$-minor-free, we define the matrix $M_{\mathbf{G}}$, whose rows are the representatives in $\mathcal{R}$ and whose columns are the graphs of $\mathcal{H}$, such that for $\mathbf{R} \in \mathcal{R}$ and $H \in \mathcal{H}$, we have $M_{\mathbf{G}}(\mathbf{R}, H) = 1$ if $\mathbf{G}$ and $\mathbf{R}$ are compatible and $H \preceq_{\mathsf{m}} \mathbf{G} \oplus \mathbf{R}$, and $M_{\mathbf{G}}(\mathbf{R}, H) = 0$ otherwise. Observe that $\mathbf{R} \in \mathcal{R}$ is the representative of $\mathbf{G}$ if and only if $M_{\mathbf{R}} = M_{\mathbf{G}}$. According to Proposition 4.4.2, $M_{\mathbf{G}}$ has size $2^{\mathcal{O}_h(t\log t)} \cdot \mathcal{O}_h(1) = 2^{\mathcal{O}_h(t\log t)}$.

For all $\mathbf{R} \in \mathcal{R}$, we compute $M_{\mathbf{R}}$. Every representative in $\mathcal{R}$ has size at most $\mathcal{O}_{s_{\mathcal{F}},h}(t)$ by Proposition 4.4.1, so when two representatives $\mathbf{R}$ and $\mathbf{R}'$ are compatible, $\mathbf{R} \oplus \mathbf{R}'$ has size $\mathcal{O}_{\ell_{\mathcal{F}},h}(t)$ as well. From [188], we know that checking if a graph $H \in \mathcal{H}$ is a minor of $\mathbf{R} \oplus \mathbf{R}'$ can be done in time $\mathcal{O}_{\ell_{\mathcal{F}},h}(t^2)$. Therefore, we can compute $M_{\mathbf{R}}$ in time $2^{\mathcal{O}_{h,s_{\mathcal{F}}}(t\log t)}$.

Let $\mathbf{G}$ be a t-boundaried graph of size k whose underlying graph is $\mathcal{F}$-minor-free. For $\mathbf{R} \in \mathcal{R}$ compatible with $\mathbf{G}$, $\mathbf{G} \oplus \mathbf{R}$ has size $\mathcal{O}_{\ell_{\mathcal{F}},h}(k+t)$, so checking if $H \in \mathcal{H}$ is a minor of $\mathbf{G} \oplus \mathbf{R}$ can be done in time $\mathcal{O}_{\ell_{\mathcal{F}},h}((k+t)^2)$. Thus we can compute $M_{\mathbf{G}}$ in time $2^{\mathcal{O}_h(t\log t)} \cdot \mathcal{O}_{\ell_{\mathcal{F}},h}((k+t)^2) = 2^{\mathcal{O}_{\ell_{\mathcal{F}},h}(t\log t+\log(k+t))}$.

Finally, we just need to find $\mathbf{R} \in \mathcal{R}$ such that $M_{\mathbf{R}} = M_{\mathbf{G}}$, which can be done in time $2^{\mathcal{O}_h(t\log t)}$. Thus, we can find the representative of $\mathbf{G}$ in time $2^{\mathcal{O}_{\ell_{\mathcal{F}},h}(t\log t+\log(k+t))}$. □

Minors of boundaried graphs. We say that a t-boundaried graph $\mathbf{G_1} = (G_1, B_1, \rho_1)$ is a minor of a t-boundaried graph $\mathbf{G_2} = (G_2, B_2, \rho_2)$, denoted by $\mathbf{G_1} \preceq_{\mathsf{m}} \mathbf{G_2}$, if there is a sequence of removals of non-boundary vertices, edge removals, and edge contractions in G_2, not allowing contractions of edges with both endpoints in B_2, that transforms G_2 to a boundaried graph that is isomorphic to G_1 (during edge contractions, boundary vertices prevail). Note that this extends the usual definition of minors in graphs without boundary.

Topological minors of boundaried graphs and folios. Let v be a vertex of degree two in a graph G. We say that a boundaried graph $\mathbf{H}$ is a *topological minor* of a boundaried graph $\mathbf{G} = (G, B, \rho)$ if $\mathbf{H}$ can be obtained from $\mathbf{G}$ after a sequence of deletion of edges of G and deletion and dissolution of vertices of $G - B$. Given $\mathbf{G} \in \mathcal{B}$ and $\ell \in \mathbb{N}$, we define the *ℓ-folio* of $\mathbf{G}$, denoted by $\ell\text{-}\mathsf{folio}(\mathbf{G})$, as the set of all boundaried graphs $\mathbf{H}$ such that $\mathbf{H}$ is a topological minor of $\mathbf{G}$ of detail at most ℓ.

4.5 Drawing on surfaces

In this section, we define Σ-*decompositions*, that are necessary to prove our structure theorem in Chapter 5, *sphere decompositions*, that are required in Chapter 10 to prove our new irrelevant vertex technique, and *renditions*, that are used to define flat walls in Subsection 4.6.2. We also provide related notions and notations.

Drawing a graph in a surface. Let Σ be a surface, possibly with boundary. A *drawing* (with crossings) in Σ is a triple $\Gamma = (U, V, E)$ such that

- V and E are finite,

- $V \subseteq U \subseteq \Sigma$,
- $V \cup \bigcup_{e \in E} e = U$ and $V \cap (\bigcup_{e \in E} e) = \emptyset$,
- for every $e \in E$, $e = h((0,1))$, where $h\colon [0,1]_{\mathbb{R}} \to U$ is a homeomorphism onto its image with $h(0), h(1) \in V$ and
- if $e, e' \in E$ are distinct, then $|e \cap e'|$ is finite.

We call the set V, sometimes denoted by $V(\Gamma)$, the *vertices of* Γ and the set E, denoted by $E(\Gamma)$, the *edges of* Γ. We also denote $U(\Gamma) = U$. If G is a graph and $\Gamma = (U, V, E)$ is a drawing with crossings in a surface Σ such that V and E naturally correspond to $V(G)$ and $E(G)$ respectively, we say that Γ is a *drawing of* G *in* Σ *(possibly with crossings).* In the case where no two edges in $E(\Gamma)$ have a common point, we say that Γ is a *drawing of* G *in* Σ *without crossings.* In this last case, the connected components of $\Sigma \setminus U$, are the *faces* of Γ. See Figure 4.2 for an illustration of a drawing Γ.

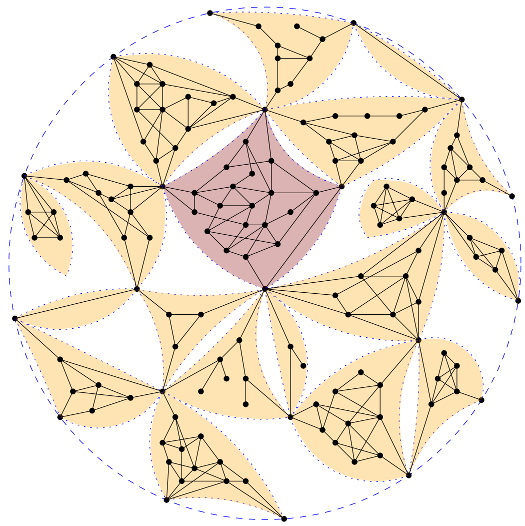

Figure 4.2: A rendition $\rho = (\Gamma, \mathcal{D})$ in the plane with one vortex (in red). Γ is the drawing in the plane composed of the black vertices and edges. $\mathcal{D}$ is the set of orange and red disks. More specifically, the red disk corresponds to a vortex cell and the orange disks are non-vortex cells.

We remind that a *closed disk* D on $\mathbb{S}^2$ is a set of points homeomorphic to the set $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ on the Euclidean plane. We use $\mathsf{bd}(D)$ to denote the boundary of D.

Σ-decompositions. Let Σ be a surface. When Σ has a boundary, then we denote it by $\mathsf{bd}(\Sigma)$. Also we refer to $\Sigma \setminus \mathsf{bd}(\Sigma)$ as the *interior* of Σ. A Σ-*decomposition* of a graph G is a pair $\delta = (\Gamma, \mathcal{D})$, where Γ is a drawing of G in Σ with crossings, and $\mathcal{D}$ is a collection of closed disks, each a subset of Σ such that

1. the disks in $\mathcal{D}$ have pairwise disjoint interiors,
2. the boundary of each disk in $\mathcal{D}$ intersects Γ in vertices only,
3. if $\Delta_1, \Delta_2 \in \mathcal{D}$ are distinct, then $\Delta_1 \cap \Delta_2 \subseteq V(\Gamma)$, and
4. every edge of Γ belongs to the interior of one of the disks in $\mathcal{D}$.

When Σ is the sphere, then we call Σ-decompositions *sphere decompositions.*

Σ-embeddings. A Σ-*embedding* of a graph G, is a Σ-decomposition $\delta = (\Gamma, \mathcal{D})$ where $\mathcal{D}$ is a collection of closed disks such that, for any disk in $\mathcal{D}$, only a single edge of Γ is drawn in its interior. For simplicity, we make the convention that, when we refer to a Σ-embedding, we just refer to the drawing of Γ, as the choice of $\mathcal{D}$ is obvious in this case. When Σ is the sphere, then we call Σ-embeddings *sphere embeddings*.

Nodes, cells, and ground vertices. For a Σ-decomposition $\delta = (\Gamma, \mathcal{D})$, let N be the set of all vertices of Γ that do not belong to the interior of the disks in $\mathcal{D}$. We refer to the elements of N as the *nodes* of δ. If $\Delta \in \mathcal{D}$, then we refer to the set $\Delta - N$ as a *cell* of δ. We denote the set of nodes of δ by $N(\delta)$ and the set of cells by $C(\delta)$. For a cell $c \in C(\delta)$, the set of nodes that belong to the closure of c is denoted by $\tilde{c}$. Given a cell $c \in C(\delta)$, we define its *disk* as $\Delta_c = \mathsf{bd}(c) \cup c$. We define $\pi_\delta \colon N(\delta) \to V(G)$ to be the mapping that assigns to every node in $N(\delta)$ the corresponding vertex of G. We also define *ground vertices* in δ as $\mathsf{ground}(\delta) = \pi_\delta(N(\delta))$. For a cell $c \in C(\delta)$ we define the graph $\sigma_\delta(c)$, or $\sigma(c)$ if δ is clear from the context, to be the subgraph of G consisting of all vertices and edges drawn in Δ_c. Note that, for any cell $c \in C(\delta)$ such that $\sigma_\delta(c)$ is not connected, we can split c into $|\mathsf{cc}(\sigma_\delta(c))|$ cells, and obtain a sphere decomposition δ' such that each cell $c \in C(\delta')$ is such that $\sigma_\delta(c)$ is connected. Hence, without loss of generality, when introducing a sphere rendition δ, we assume $\sigma_\delta(c)$ to be connected for each $c \in C(\delta)$.

Vortices. Let G be a graph, Σ be a surface, and $\delta = (\Gamma, \mathcal{D})$ be a Σ-decomposition of G. A cell $c \in C(\delta)$ is called a *vortex* if $|\tilde{c}| \geq 4$. Moreover, we call δ *vortex-free* if no cell in $C(\delta)$ is a vortex.

Societies. Let Ω be a cyclic permutation of the elements of some set which we denote by $V(\Omega)$. A *society* is a pair (G, Ω), where G is a graph and Ω is a cyclic permutation with $V(\Omega) \subseteq V(G)$.

Crosses. A *cross* in a society (G, Ω) is a pair (P_1, P_2) of disjoint paths[1] in G such that P_i has endpoints $s_i, t_i \in V(\Omega)$ and is otherwise disjoint from $V(\Omega)$, and the vertices s_1, s_2, t_1, t_2 occur in Ω in the order listed.

Renditions. Let (G, Ω) be a society and let Δ be a closed disk in Σ. A *rendition* in Σ of (G, Ω) is a Σ-decomposition ρ of G such that $\pi_\rho(N(\rho) \cap \mathsf{bd}(\Delta)) = V(\Omega)$, mapping one of the two cyclic orders (clockwise or counterclockwise) of $\mathsf{bd}(\Delta)$ to the order of Ω. See Figure 4.2 for an illustration.

Rural societies. A society is *rural* if it has a vortex-free rendition.

Proposition 4.5.1 ([194, 265]). *A society (G, Ω) in the disk has no cross if and only if it is rural.*

Deleting a set. Given a set $X \subseteq V(G)$, we denote by $\delta - X$ the sphere decomposition $\delta' = (\Gamma', \mathcal{D}')$ of $G - X$ where Γ' is obtained by the drawing Γ after removing all points in X and all drawings of edges with an endpoint in X. For every point $x \in \pi_\delta^{-1}(X \cap N(\delta))$, we pick Δ_x to be an open disk containing x and not containing any point of some remaining vertex or edge and such that no two such disks intersect. We also set $\Delta_X = \bigcup_{x \in \pi_\delta^{-1}(X)} \Delta_x$ and we define $\mathcal{D}' = \{D \setminus \Delta_X \mid D \in \mathcal{D}\}$. Clearly, there is a one to one correspondence between the cells of δ and the cells of δ'. If a cell c of δ corresponds to a cell c' of δ', then we call c' the *heir* of c in δ' and we call c the *precursor* of c' in δ.

[1] When we say two paths are *disjoint* we mean that their vertex sets are disjoint.

Grounded graphs. Let δ be a Σ-decomposition of a graph G in a surface Σ. We say that a cycle C of G is *grounded in* δ if C uses edges of $\sigma(c_1)$ and $\sigma(c_2)$ for two distinct cells $c_1, c_2 \in C(\delta)$. A 2-connected subgraph H of G is said to be *grounded in* δ if every cycle in H is grounded in δ.

Tracks. Let δ be a Σ-decomposition of a graph G in a surface Σ. For every cell $c \in C(\delta)$ with $|\tilde{c}| = 2$ we select one of the components of $\mathsf{bd}(c) - \tilde{c}$. This selection is called a *tie-breaker* in δ, and we assume every Σ-decomposition to come equipped with a tie-breaker. Let C be a cycle grounded in δ. We define the *track* of C as follows Let $P_1, \ldots, P_k$ be distinct maximal subpaths of C such that P_i is a subgraph of $\sigma(c)$ for some cell c. Fix an index i. The maximality of P_i implies that its endpoints are $\pi_\delta(n_1)$ and $\pi_\delta(n_2)$ for distinct δ-nodes $n_1, n_2 \in N(\delta)$. If $|\tilde{c}| = 2$, define L_i to be the component of $\mathsf{bd}(c) - \{n_1, n_2\}$ selected by the tie-breaker, and if $|\tilde{c}| = 3$, define L_i to be the component of $\mathsf{bd}(c) - \{n_1, n_2\}$ that is disjoint from $\tilde{c}$. Finally, we define L'_i by slightly pushing L_i to make it disjoint from all cells in $C(\delta)$. We define such a curve L'_i for all i while ensuring that the curves intersect only at a common endpoint. The *track* of C is defined to be $\bigcup_{i \in [k]} L'_i$. So the track of a cycle is the homeomorphic image of the unit circle.

δ-aligned disks. We say a closed disk Δ in Σ is *δ-aligned* if its boundary intersects Γ only in nodes of δ. We denote by Ω_Δ one of the cyclic orderings of the vertices on the boundary of Δ. We define the *inner graph* of a δ-aligned closed disk Δ as

$$\mathsf{inner}_\delta(\Delta) \coloneqq \bigcup_{c \in C(\delta) \text{ and } c \subseteq \Delta} \sigma(c)$$

and the *outer graph of* Δ as

$$\mathsf{outer}_\delta(\Delta) \coloneqq \bigcup_{c \in C(\delta) \text{ and } c \cap \Delta \subseteq \mathsf{ground}(\delta)} \sigma(c).$$

If Δ is δ-aligned, we define $\Gamma \cap \Delta$ to be the drawing of $\mathsf{inner}_\delta(\Delta)$ in Δ which is the restriction of Γ in Δ. If moreover $|\Omega_\Delta| \geq 4$, we denote by $\delta[\Delta]$ the rendition $(\Gamma \cap \Delta, \{\Delta_c \in \mathcal{D} \mid c \subseteq \Delta\})$ of $(\mathsf{inner}_\delta(\Delta), \Omega_\Delta)$ in Δ.

Let $\delta = (\Gamma, \mathcal{D})$ be a Σ-decomposition of a graph G in a surface Σ. Let C be a cycle in G that is grounded in δ, such that the track T of C bounds a closed disk Δ_C in Σ. We define the *outer* (resp. *inner*) *graph* of C in δ as the graph $\mathsf{outer}_\delta(C) \coloneqq \mathsf{outer}_\delta(\Delta_C)$ (resp. $\mathsf{inner}_\delta(C) \coloneqq \mathsf{inner}_\delta(\Delta_C)$).

Paths. If P is a path and x and y are vertices on P, we denote by xPy the subpath of P with endpoints x and y. Moreover, if s and t are the endpoints of P, and we order the vertices of P by traversing P from s to t, then xP denotes the path xPt and Px denotes the path sPx. Let P be a path from s to t and Q be a path from q to p. If x is a vertex in $V(P) \cap V(Q)$ such that Px and xQ intersect only in x, then PxQ is the path obtained from the union of Px and xQ. Let $X, Y \subseteq V(G)$. A path is an *X-Y-path* if it has one endpoint in X and the other in Y and is internally disjoint from $X \cup Y$, Whenever we consider X-Y-paths we implicitly assume them to be ordered starting in X and ending in Y, except if stated otherwise. An *X-path* is an X-X-path of length at least one. In a society (G, Ω), we write Ω-path as a shorthand for a $V(\Omega)$-path.

Segments. Let (G, Ω) be a society. A *segment* of Ω is a set $S \subseteq V(\Omega)$ such that there do not exist $s_1, s_2 \in S$ and $t_1, t_2 \in V(\Omega) \setminus S$ such that s_1, t_1, s_2, t_2 occur in Ω in the order listed. A vertex $s \in S$ is an *endpoint* of the segment S if there is a vertex $t \in V(\Omega) \setminus S$ which immediately precedes or immediately succeeds s in the order Ω. For vertices $s, t \in V(\Omega)$, if t immediately precedes s, we define $s\Omega t$ to be the *trivial segment* $V(\Omega)$, and otherwise we define $s\Omega t$ to be the uniquely determined segment with first vertex s and last vertex t.

4.6 Flat walls

In this section, we define formally walls and related notions, that are essential to find an irrelevant vertex (cf. Subsection 3.1.3), using the framework of [286]. More precisely, in Subsection 4.6.1, we introduce walls and several notions concerning them. Using the above notions, in Subsection 4.6.2, we define flat walls and provide some results about them, including two versions of the Flat Wall theorem. In Subsection 4.6.3, we define canonical partitions, that essentially express how to contract a wall to obtain a grid and are necessary to find an obligatory set (cf. Subsection 3.1.5). Finally, we define in Subsection 4.6.4 homogeneous flat walls and in Subsection 4.6.5 tight flat walls, which are necessary to find an irrelevant vertex.

These definitions and results are used in Chapter 5, Chapter 7, Chapter 8, and Chapter 10.

4.6.1 Walls and subwalls

We start with some basic definitions about walls.

Grids. Let $k, r \in \mathbb{N}$. The $(k \times r)$-*grid*, denoted by $\Gamma_{k,r}$, is the graph whose vertex set is $[k] \times [r]$ and two vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| + |j - j'| = 1$. See Figure 1.19 for an illustration. We call the path where vertices appear as $(i, 1), (i, 2), \ldots, (i, r)$ the i-*th row* and the path where vertices appear as $(1, j), (2, j), \ldots, (k, j)$ the j-*th column* of the grid.

Central grids. Let $k, r \in \mathbb{N}_{\geq 2}$. We define the *perimeter* of a $(k \times r)$-grid to be the unique cycle of the grid of length at least three that does not contain vertices of degree four. We shorten the notation $(r \times r)$-grid as an r-*grid*.

Let $r \in \mathbb{N}_{\geq 2}$ and Γ_r be an r-grid. Given an $i \in \lceil \frac{r}{2} \rceil$, we define the i-*th layer* of Γ_r recursively as follows. The first layer of Γ_r is its perimeter, while, if $i \geq 2$, the i-th layer of Γ_r is the $(i-1)$-th layer of the grid created if we remove from Γ_r its perimeter. Given two odd integers $q, r \in \mathbb{N}_{\geq 3}$ such that $q \leq r$ and an r-grid Γ_r, we define the *central* q-*grid* of Γ_r to be the q-grid obtained from Γ_r if we remove from Γ_r its $\frac{r-q}{2}$ first layers.

Elementary walls. An *elementary* (k, ℓ)-*wall* $W_{k,\ell}$ for $k, \ell \geq 3$, is obtained from the $(k \times 2\ell)$-grid $\Gamma_{k,2\ell}$ by deleting every odd edge in every odd column and every even edge in every even column, and then deleting all degree-one vertices. The *rows* of $W_{k,\ell}$ are the subgraphs of $W_{k,\ell}$ induced by the rows (or *horizontal paths*) of $\Gamma_{k,2\ell}$, while the j*th column* (or j-*th vertical path*) of $W_{k,\ell}$ is the subgraph induced by the vertices of columns $2j - 1$ and $2j$ of $\Gamma_{k,2\ell}$. We define the *perimeter* of $W_{k,\ell}$ to be the subgraph induced by $\{(i, j) \in V(W_{k,\ell}) \mid j \in \{1, 2, 2\ell, 2\ell - 1\}$ and $i \in [k]$, or $i \in \{1, k\}$ and $j \in [2\ell]\}$.

Walls. A (k, ℓ)-*wall* W is a graph isomorphic to a subdivision of $W_{k,\ell}$. The vertices of degree three in W are called the *3-branch vertices*. In other words, W is obtained from a graph W' isomorphic to $W_{k,\ell}$ by subdividing each edge of W' an arbitrary (possibly zero) number of times. We define rows and columns of (k, ℓ)-walls analogously to their definition for elementary walls. A *(elementary)* k-*wall* W is a (elementary) (k, k)-wall (see Figure 4.3) and we refer to k as the *height* of W. Note that we will prefer to choose k odd for symmetry. A *wall* is a (k, ℓ)-wall for some k, ℓ. The vertices in the perimeter of an elementary r-wall that have degree two are called *pegs*, while the vertices $(1,1),(2,r),(2r-1,1),(2r,r)$ are called *corners* (notice that the corners are also pegs).

An h-wall W' is a *subwall* of some k-wall W where $h \leq k$ if every row (column) of W' is contained in a row (column) of W.

Notice that, as $k \geq 3$, an elementary k-wall is a planar graph that has a unique (up to topological isomorphism) embedding in the plane $\mathbb{R}^2$ such that all its finite faces are incident to exactly six edges. The perimeter of an elementary r-wall is the cycle bounding its infinite face, while the cycles bounding its finite faces are called *bricks*. A cycle of a wall W, obtained from the elementary wall W', is the *perimeter* of W, denoted by $D(W)$, if its 3-branch vertices are the vertices of the perimeter of W'. A brick of W is *internal* if it is disjoint from $D(W)$.

We present the following result of Kawarabayashi and Kobayashi [187], which provides a linear relation between the treewidth and the height of a largest wall in a minor-free graph.

Proposition 4.6.1 ([187]). *There is a function $f_{4.6.1}: \mathbb{N} \to \mathbb{N}$ such that, for every $t, r \in \mathbb{N}$ and every graph G that does not contain K_t as a minor, if $\mathsf{tw}(G) \geq f_{4.6.1}(t) \cdot r$, then G contains an r-wall as a subgraph. In particular, one may choose $f_{4.6.1}(t) = 2^{\mathcal{O}(t^2 \cdot \log t)}$.*

Layers. The *layers* of an r-wall W are recursively defined as follows. The first layer of W is its perimeter. For $i = 2, \ldots, (r-1)/2$, the i-th layer of W is the $(i-1)$-th layer of the subwall W' obtained from W after removing from W its perimeter and removing recursively all occurring vertices of degree one. We refer to the $(r-1)/2$-th layer as the *inner layer* of W. The *central vertices* of an r-wall are its two 3-branch vertices that do not belong to any of its layers and that are connected by a path of W that does not intersect any layer. See Figure 4.3 for an illustration of the notions defined above.

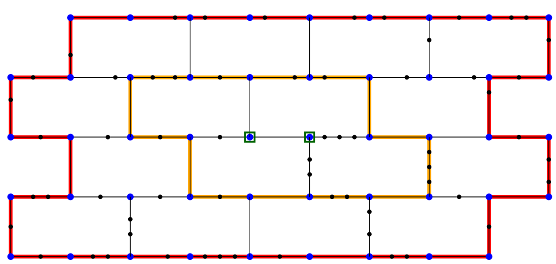

Figure 4.3: A 5-wall. Its first layer is depicted in red and its second layer in orange. Its central vertices are depicted in a green square.

Central walls. Given an r-wall W and an odd $q \in \mathbb{N}_{\geq 3}$ where $q \leq r$, we define the *central q-subwall* of W, denoted by $W^{(q)}$, to be the q-wall obtained from W after removing its first $(r-q)/2$ layers and all occurring vertices of degree one.

Tilts. The *interior* of a wall W is the graph obtained from W if we remove from it all edges of $D(W)$ and all vertices of $D(W)$ that have degree two in W. Given two walls W and $\tilde{W}$ of a graph G, we say that $\tilde{W}$ is a *tilt* of W if $\tilde{W}$ and W have identical interiors.

Minor models grasped by walls. Let G be a graph and W be an r-wall in G. Let $P_1, \ldots, P_r$ be the horizontal paths and $Q_1, \ldots, Q_r$ be the vertical paths of W. Let $t \in \mathbb{N}_{\geq 1}$. A model $\{S_v \mid v \in V(K_t)\}$ of K_t in G is *grasped* by W if, for all $v \in V(K_t)$, there exist $(i,j) \in [r]^2$ such that $V(P_i) \cap V(Q_j) \subseteq S_v$.

4.6.2 Flatness pairs

In this subsection we define the notion of a flat wall. In order for the formal statements of this section to be mathematically correct, we need to introduce a number of notions originating in [286]. We would like to stress that these notions are needed for the formal statements of the results, but that most of them are not fundamental for the main conceptual contributions of this thesis. We refer the reader to [286] for a more detailed exposition of these definitions and the reasons for which they were introduced.

Flat walls. Let G be a graph and let W be an r-wall of G, for some odd integer $r \geq 3$. We say that a pair $(P, C) \subseteq V(D(W)) \times V(D(W))$ is a *choice of pegs and corners for* W if W is a subdivision of an elementary r-wall $\bar{W}$ where P and C are the pegs and the corners of $\bar{W}$, respectively (clearly, $C \subseteq P$). To get more intuition, notice that a wall W can occur in several ways from the elementary wall $\bar{W}$, depending on the way the vertices in the perimeter of $\bar{W}$ are subdivided. Each of them gives a different selection (P, C) of pegs and corners of W.

We say that W is a *flat* r*-wall* of G if there is a separation (X, Y) of G and a choice (P, C) of pegs and corners for W such that:

- $V(W) \subseteq Y$,
- $P \subseteq X \cap Y \subseteq V(D(W))$, and
- if Ω is the cyclic ordering of the vertices $X \cap Y$ as they appear in $D(W)$, then there exists a vortex-free rendition ρ of $(G[Y], \Omega)$ in the sphere.

We say that W is a *flat wall* of G if it is a flat r-wall for some odd integer $r \geq 3$.

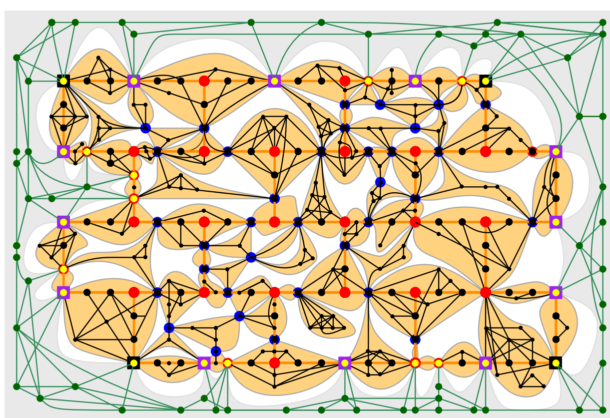

Figure 4.4: Illustration of a flatness pair $(W, \mathfrak{R})$ of a graph G (adapted from [24, Figure 4]). The edges of W are depicted in orange and the $\mathcal{R}$-compass of W is the union of all parts of G that are drawn in light orange cells. The yellow vertices are the vertices of $V(\Omega)$ and the squared vertices are the choice of pegs (in purple) and corners (in black) of W.

Flatness pairs. Given the above, we say that the choice of the 5-tuple $\mathfrak{R} = (X, Y, P, C, \rho)$ *certifies that* W *is a flat wall of* G. We call the pair $(W, \mathfrak{R})$ a *flatness pair* of G and define the *height* of the pair $(W, \mathfrak{R})$ to be the height of W. See Figure 4.4 for an illustration.

We call the graph $G[Y]$ the $\mathfrak{R}$-*compass* of W in G, denoted by $\mathsf{Compass}_{\mathfrak{R}}(W)$. We can assume that $\mathsf{Compass}_{\mathfrak{R}}(W)$ is connected, updating $\mathfrak{R}$ by removing from Y the vertices of all the connected components of $\mathsf{Compass}_{\mathfrak{R}}(W)$ except for the one that contains W and including them in X (Γ can also be easily modified according to the removal of the aforementioned vertices from Y). We define the *flaps* of the wall W in $\mathfrak{R}$ as $\mathsf{Flaps}_{\mathfrak{R}}(W) := \{\sigma(c) \mid c \in C(\rho)\}$. Given a flap $F \in \mathsf{Flaps}_{\mathfrak{R}}(W)$, we define its *base* as $\partial F := V(F) \cap \pi(N(\Gamma))$.

The Flat Wall theorem is a result of Robertson and Seymour [271] that says that a graph G either contains a big clique as a minor, or has bounded treewidth, or contains a flat wall. Many versions of the Flat Wall theorem were proved over time, including the following one.

Proposition 4.6.2 ([194]). *There are two functions $f_{4.6.2}, g_{4.6.2} : \mathbb{N} \to \mathbb{N}$, such that the images of $f_{4.6.2}$ are odd integers, and an algorithm with the following specifications:*

$\texttt{Grasped-or-Flat}(G, r, t, W)$
***Input**: A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, $t \in \mathbb{N}_{\geq 1}$, and an $f_{4.6.2}(t) \cdot r$-wall W of G.*
***Output**: One of the following:*

- *Either a model of a K_t-minor in G grasped by W, or*
- *a set $A \subseteq V(G)$ of size at most $g_{4.6.2}(t)$ and a flatness pair $(W', \mathfrak{R}')$ of $G - A$ of heigth r such that W' is a tilt of some subwall $\tilde{W}'$ of W.*

Moreover, $f_{4.6.2}(t) = \mathcal{O}(t^{26})$, $g_{4.6.2}(t) = \mathcal{O}(t^{24})$, and the algorithm runs in time $\mathcal{O}(t^{24} m + n)$.

Cell classification. Given a cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$, we say that C is $\mathfrak{R}$-*normal* if it is *not* a subgraph of a flap $F \in \mathsf{Flaps}_{\mathfrak{R}}(W)$. Given an $\mathfrak{R}$-normal cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$, we call a cell c of $\mathfrak{R}$ C-*perimetric* if $\sigma(c)$ contains some edge of C. Notice that if c is C-perimetric, then $\pi_\rho(\tilde{c})$ contains two points $p, q \in N(\rho)$ such that $\pi_\rho(p)$ and $\pi_\rho(q)$ are vertices of C where one, say P_c^{in}, of the two $(\pi_\rho(p), \pi_\rho(q))$-subpaths of C is a subgraph of $\sigma(c)$ and the other, denoted by P_c^{out}, $(\pi_\rho(p), \pi_\rho(q))$-subpath contains at most one internal vertex of $\sigma(c)$, which should be the (unique) vertex z in $\pi_\rho(\tilde{c}) \setminus \{\pi_\rho(p), \pi_\rho(q)\}$. We pick a (p, q)-arc A_c in $\hat{c} := c \cup \tilde{c}$ such that $\pi_\rho^{-1}(z) \in A_c$ if and only if P_c^{in} contains the vertex z as an internal vertex.

We consider the circle $K_C = \bigcup\{A_c \mid c \text{ is a } C\text{-perimetric cell of } \mathfrak{R}\}$ and we denote by Δ_C the closed disk bounded by K_C that is contained in Δ. A cell c of $\mathfrak{R}$ is called C-*internal* if $c \subseteq \Delta_C$ and is called C-*external* if $\Delta_C \cap c = \emptyset$. Notice that the cells of $\mathfrak{R}$ are partitioned into C-internal, C-perimetric, and C-external cells.

A cell c of $\mathfrak{R}$ is *untidy* if $\pi_\rho(\tilde{c})$ contains a vertex x of W such that two of the edges of W that are incident to x are edges of $\sigma(c)$. Notice that if c is untidy then $|\tilde{c}| = 3$. A cell c of $\mathfrak{R}$ is *tidy* if it is not untidy.

Let c be a tidy C-perimetric cell of $\mathfrak{R}$ where $|\tilde{c}| = 3$. Notice that $c \setminus A_c$ has two arcwise-connected components and one of them is an open disk D_c that is a subset of Δ_C. If the closure $\overline{D}_c$ of D_c contains only two points of $\tilde{c}$ then we call the cell c C-*marginal*.

Influence. For every $\mathfrak{R}$-normal cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$ we define the set $\mathsf{influence}_{\mathfrak{R}}(C) = \{\sigma(c) \mid c \text{ is a cell of } \mathfrak{R} \text{ that is not } C\text{-external}\}$.

A wall W' of $\mathsf{Compass}_{\mathfrak{R}}(W)$ is $\mathfrak{R}$-*normal* if $D(W')$ is $\mathfrak{R}$-normal. Notice that every wall of W (and hence every subwall of W) is an $\mathfrak{R}$-normal wall of $\mathsf{Compass}_{\mathfrak{R}}(W)$. We denote by $\mathcal{S}_{\mathfrak{R}}(W)$ the set of all $\mathfrak{R}$-normal walls of $\mathsf{Compass}_{\mathfrak{R}}(W)$. Given a wall $W' \in \mathcal{S}_{\mathfrak{R}}(W)$ and a cell c of $\mathfrak{R}$, we say that c is W'-*perimetric/internal/external/marginal* if c is $D(W')$-perimetric/internal/external/marginal, respectively. We also use $K_{W'}$, $\Delta_{W'}$, $\mathsf{influence}_{\mathfrak{R}}(W')$ as shortcuts for $K_{D(W')}$, $\Delta_{D(W')}$, $\mathsf{influence}_{\mathfrak{R}}(D(W'))$, respectively.

Regular flatness pairs. We call a flatness pair $(W, \mathfrak{R})$ of a graph G *regular* if none of its cells is W-external, W-marginal, or untidy.

The next result is another version of the Flat Wall theorem. Compared to Proposition 4.6.2, in Proposition 4.6.3, we lose the fact that the clique-minor is grasped by the wall given in the input, but we gain the fact that the compass of the flat wall has bounded treewidth.

Proposition 4.6.3 ([286]). *There exist a function $f_{4.6.3} : \mathbb{N} \to \mathbb{N}$ and an algorithm with the following specifications:*

`Clique-Or-twFlat`(G, r, t)
Input: *A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, and $t \in \mathbb{N}_{\geq 1}$.*
Output: *One of the following:*

- *Either a report that K_t is a minor of G, or*
- *a tree decomposition of G of width at most $f_{4.6.3}(t) \cdot r$, or*
- *a set $A \subseteq V(G)$ of size at most $g_{4.6.2}(t)$ and a regular flatness pair $(W', \mathfrak{R}')$ of $G - A$ of height r whose $\mathfrak{R}'$-compass has treewidth at most $f_{4.6.3}(t) \cdot r$.*

Moreover, $f_{4.6.3}(t) = 2^{\mathcal{O}(t^2 \log t)}$ and this algorithm runs in time $2^{\mathcal{O}_t(r^2)} \cdot n$. The algorithm can be modified to obtain an explicit dependence on t in the running time, namely $2^{2^{\mathcal{O}(t^2 \log t)} \cdot r^3 \log r} \cdot n$.

Tilts of flatness pairs. Let $(W, \mathfrak{R})$ and $(\tilde{W}', \tilde{\mathfrak{R}}')$ be two flatness pairs of a graph G and let $W' \in \mathcal{S}_{\mathfrak{R}}(W)$. We assume that $\mathfrak{R} = (X, Y, P, C, \rho)$ and $\tilde{\mathfrak{R}}' = (X', Y', P', C', \rho')$. We say that $(\tilde{W}', \tilde{\mathfrak{R}}')$ is a W'-*tilt* of $(W, \mathfrak{R})$ if

- $\tilde{\mathfrak{R}}'$ does not have $\tilde{W}'$-external cells,
- $\tilde{W}'$ is a tilt of W',
- the set of $\tilde{W}'$-internal cells of $\tilde{\mathfrak{R}}'$ is the same as the set of W'-internal cells of $\mathfrak{R}$ and their images via $\sigma_{\rho'}$ and σ_ρ are also the same,
- $\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')$ is a subgraph of $\bigcup \mathsf{influence}_{\mathfrak{R}}(W')$, and
- if c is a cell in $C(\Gamma') \setminus C(\Gamma)$, then $|\tilde{c}| \leq 2$.

The next observation follows from the third item above and the fact that the cells corresponding to flaps containing a central vertex of W' are all internal (recall that the height of a wall is always at least three).

Observation 4.6.4. *Let $(W, \mathfrak{R})$ be a flatness pair of a graph G and $W' \in \mathcal{S}_{\mathfrak{R}}(W)$. For every W'-tilt $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $(W, \mathfrak{R})$, the central vertices of W' belong to the vertex set of $\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')$.*

Also, given a regular flatness pair $(W, \mathfrak{R})$ of a graph G and a $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, for every W'-tilt $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $(W, \mathfrak{R})$, by definition none of its cells is $\tilde{W}'$-external, $\tilde{W}'$-marginal, or untidy – thus, $(\tilde{W}', \tilde{\mathfrak{R}}')$ is regular. Therefore, regularity of a flatness pair is a property that its tilts "inherit".

Observation 4.6.5. *If $(W, \mathfrak{R})$ is a regular flatness pair, then for every $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, every W'-tilt of $(W, \mathfrak{R})$ is also regular.*

Furthermore, we need the following propositions, that are the main results of [286].

Proposition 4.6.6 ([286]). *There exists an algorithm that, given a graph G, a flatness pair $(W, \mathfrak{R})$ of G, and a wall $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, outputs a W'-tilt of $(W, \mathfrak{R})$ in time $\mathcal{O}(n+m)$.*

Proposition 4.6.7 ([286]). *There exists an algorithm that, given a graph G and a flatness pair $(W, \mathfrak{R})$ of G, outputs in time $\mathcal{O}(n+m)$ a regular flatness pair $(W^*, \mathfrak{R}^*)$ of G with the same weight as $(W, \mathfrak{R})$ such that $\mathsf{Compass}_{\mathfrak{R}^*}(W^*) \subseteq \mathsf{Compass}_{\mathfrak{R}}(W)$.*

4.6.3 Canonical partitions

In this subsection, we define the notion of canonical partition of a graph G with respect to some wall W of G. This refers to a partition of the vertex set of G into bags that follow the structure of W. Essentially, the goal is to be able to contract each of these bags to obtain a grid that is a minor of W and thus of G. In particular, we prove in Section 7.5 that if G contains as a minor a grid Γ along with a set A whose vertices have sufficiently many neighbors in the grid, then some vertex in A is obligatory. We use canonical partitions here to easily find such a structure given a wall of G. We also use such a result in Section 8.4.

For this reason, we start by defining the canonical partition of a wall, as a "canonical" way to partition the vertices of the wall into connected subsets that preserve the grid-like structure of the wall.

Canonical partition of a wall. Let $r \geq 3$ be an odd integer. Let W be an r-wall and let $P_1, \ldots, P_r$ (resp. $L_1, \ldots, L_r$) be its vertical (resp. horizontal) paths. For every even (resp. odd) $i \in [2, r-1]$ and every $j \in [2, r-1]$, we define $A^{(i,j)}$ to be the subpath of P_i that starts from a vertex of $P_i \cap L_j$ and finishes at a neighbor of a vertex in L_{j+1} (resp. L_{j-1}), such that $P_i \cap L_j \subseteq A^{(i,j)}$ and $A^{(i,j)}$ does not intersect L_{j+1} (resp. L_{j-1}). Similarly, for every $i, j \in [2, r-1]$, we define $B^{(i,j)}$ to be the subpath of L_j that starts from a vertex of $P_i \cap L_j$ and finishes at a neighbor of a vertex in P_{i-1}, such that $P_i \cap L_j \subseteq B^{(i,j)}$ and $B^{(i,j)}$ does not intersect P_{i-1}.

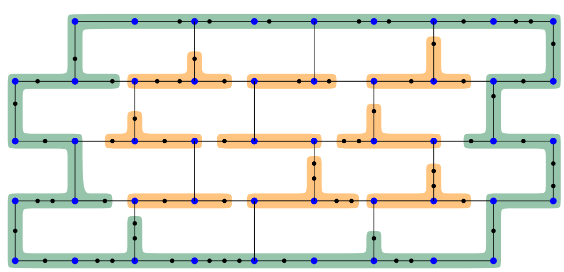

Figure 4.5: A 5-wall and its canonical partition $\mathcal{Q}$. The green bag is the external bag Q_{ext} and the orange bags are the internal bags of $\mathcal{Q}$. Observe that if we contract each internal bag of $\mathcal{Q}$, then we obtain a (3×3)-grid.

For every $i, j \in [2, r-1]$, we denote by $Q^{(i,j)}$ the graph $A^{(i,j)} \cup B^{(i,j)}$ and by Q_{ext} the graph $W - \bigcup_{i,j \in [2,r-1]} V(Q_{i,j})$. Now consider the collection $\mathcal{Q} = \{Q_{\text{ext}}\} \cup \{Q_{i,j} \mid i, j \in [2, r-1]\}$ and observe that the graphs in $\mathcal{Q}$ are connected subgraphs of W and their vertex sets form a partition of $V(W)$. We call $\mathcal{Q}$ the *canonical partition* of W. Also, we call every $Q_{i,j}, i, j \in [2, r-1]$ an *internal bag* of $\mathcal{Q}$, while we refer to Q_{ext} as the *external bag* of $\mathcal{Q}$. See Figure 4.5 for an illustration of the notions defined above. For every $i \in [(r-1)/2]$, we say that a set $Q \in \mathcal{Q}$ is an *i-internal bag of $\mathcal{Q}$* if $V(Q)$ does not contain any vertex of the first i layers of W. Notice that the 1-internal bags of $\mathcal{Q}$ are the internal bags of $\mathcal{Q}$.

Canonical partitions of a graph with respect to a wall. Let W be a wall of a graph G. Consider the canonical partition $\mathcal{Q}$ of W. The *enhancement* of the canonical partition $\mathcal{Q}$ on G is the following operation. We set $\tilde{\mathcal{Q}} := \mathcal{Q}$ and, as long as there is a vertex $x \in G - V(\bigcup \tilde{\mathcal{Q}})$ that is adjacent to a vertex of a graph $Q \in \tilde{\mathcal{Q}}$, we update $\tilde{\mathcal{Q}} := \tilde{\mathcal{Q}} \setminus \{Q\} \cup \{\tilde{Q}\}$, where $\tilde{Q} = G[\{x\} \cup V(Q)]$. We call the $\tilde{Q} \in \tilde{\mathcal{Q}}$ that contains Q_{ext} as a subgraph the *external bag* of $\tilde{\mathcal{Q}}$, and we denote it by $\tilde{Q}_{\text{ext}}$, while we call *internal bags* of $\tilde{\mathcal{Q}}$ all graphs in $\tilde{\mathcal{Q}} \setminus \{\tilde{Q}_{\text{ext}}\}$. Moreover, we enhance $\tilde{\mathcal{Q}}$ by adding all vertices of $G - \bigcup_{\tilde{Q} \in \tilde{\mathcal{Q}}} V(\tilde{Q})$ to its external bag, i.e., by updating $\tilde{Q}_{\text{ext}} := G[V(\tilde{Q}_{\text{ext}}) \cup V(G) \setminus \bigcup_{\tilde{Q} \in \tilde{\mathcal{Q}}} V(\tilde{Q})]$.

We call such a partition $\tilde{\mathcal{Q}}$ a *W-canonical partition of G*. Notice that a W-canonical partition of G is not unique, since the sets in $\mathcal{Q}$ can be "expanded" arbitrarily when introducing vertex x. However, the canonical partition $\mathcal{Q}$ of W is unique.

Let W be an r-wall of a graph G, for some odd integer $r \geq 3$ and let $\tilde{\mathcal{Q}}$ be a W-canonical partition of G. For every $i \in [(r-1)/2]$, we say that a set $Q \in \tilde{\mathcal{Q}}$ is an *i-internal bag of $\tilde{\mathcal{Q}}$* if it contains an i-internal bag of $\mathcal{Q}$ as a subgraph.

The next result is proved in [285] and intuitively states that, given a flatness pair $(W, \mathfrak{R})$ of big enough height and a W-canonical partition $\tilde{\mathcal{Q}}$ of G, we can find a packing of subwalls of W that are inside some central part of W and such that the vertex set of every internal bag of $\tilde{\mathcal{Q}}$ intersects the vertices of the flaps in the influence of at most one of these walls.

Proposition 4.6.8 ([285]). *There exist a function $f_{4.6.8} : \mathbb{N}^3 \to \mathbb{N}$ and an algorithm with the following specifications:*

$\texttt{Packing}(l, r, p, G, W, \mathfrak{R}, \tilde{\mathcal{Q}})$
Input: *Integers $l, r, p \in \mathbb{N}_{\geq 1}$, where $r \geq 3$ is odd, a graph G, and a flatness pair $(W, \mathfrak{R})$ of G of height at least $f_{4.6.8}(l, r, p)$.*
Output: *A collection $\mathcal{W} = \{W^1, \ldots, W^l\}$ of r-subwalls of W such that, for every W-canonical partition $\tilde{\mathcal{Q}}$ of G,*

- *for every $i \in [l]$, $\bigcup \mathsf{influence}_{\mathfrak{R}}(W^i)$ is a subgraph of $\bigcup \{Q \mid Q$ is a p-internal bag of $\tilde{\mathcal{Q}}\}$ and*
- *for every $i, j \in [l]$, with $i \neq j$, there is no internal bag of $\tilde{\mathcal{Q}}$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}}(W^j))$.*

Moreover, $f_{4.6.8}(l, r, p) = \mathcal{O}(\sqrt{l} \cdot r + p)$ and the algorithm runs in time $\mathcal{O}(n + m)$.

4.6.4 Homogeneous walls

In this subsection, we define homogeneous flat walls. Intuitively, homogeneous flat walls are flat walls where each brick has the same "color", where a "color" express what kind of topological minor can be routed in the (augmented) flaps of the brick. This essentially implies that a topological minor, and by extension a minor, can be routed similarly through any brick of a homogeneous flat wall.

Augmented flaps. Let G be a graph, A be a subset of $V(G)$ of size a, and $(W, \mathfrak{R})$ be a flatness pair of $G - A$. For each flap $F \in \mathsf{Flaps}_{\mathfrak{R}}(W)$, we consider a labeling $\ell_F : \partial F \to \{1, 2, 3\}$ such that the set of labels assigned by ℓ_F to ∂F is one of $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$. We consider a bijection $\rho_A : A \to [a]$. The labelings in $\mathcal{L} = \{\ell_F \mid F \in \mathsf{Flaps}_{\mathfrak{R}}(W)\}$ and the labeling ρ_A will be useful for defining a set of boundaried graphs that we will call augmented flaps. We first need some more definitions.

Given a flap $F \in \mathsf{Flaps}_{\mathfrak{R}}(W)$, we define an ordering $\Omega(F) = (x_1, \ldots, x_q)$, with $q \leq 3$, of the vertices of ∂F so that

- $(x_1, \ldots, x_q)$ is a counter-clockwise cyclic ordering of the vertices of ∂F as they appear in the corresponding cell of $C(\Gamma)$. Notice that this cyclic ordering is significant only when $|\partial F| = 3$, in the sense that (x_1, x_2, x_3) remains invariant under shifting, i.e., (x_1, x_2, x_3) is the same as (x_2, x_3, x_1) but not under inversion, i.e., (x_1, x_2, x_3) is not the same as (x_3, x_2, x_1), and
- for $i \in [q]$, $\ell_F(x_i) = i$.

Notice that the second condition is necessary for completing the definition of the ordering $\Omega(F)$, and this is the reason why we set up the labelings in $\mathcal{L}$.

For each $F \in \mathsf{Flaps}_{\mathfrak{R}}(W)$ with $t_F := |\partial F|$, we fix $\rho_F : \partial F \to [a+1, a+t_F]$ such that $(\rho_F^{-1}(a+1), \ldots, \rho_F^{-1}(a+t_F)) = \Omega(F)$. Also, we define the boundaried graph

$$\mathbf{F}^A := (G[A \cup F], A \cup \partial F, \rho_A \cup \rho_F)$$

and we denote by F^A the underlying graph of $\mathbf{F}^A$. We call $\mathbf{F}^A$ an *A-augmented flap* of the flatness pair $(W, \mathfrak{R})$ of $G - A$ in G.

Palettes and homogeneity. For each $\mathfrak{R}$-normal cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$, we define $(A, \ell)\text{-}\mathsf{palette}(C) = \{\ell\text{-}\mathsf{folio}(\mathbf{F}^A) \mid F \in \mathsf{influence}_{\mathfrak{R}}(C)\}$. We say that the flatness pair $(W, \mathfrak{R})$ of $G - A$ is *ℓ-homogeneous with respect to A* if every *internal* brick of W has the *same* (A, ℓ)-$\mathsf{palette}$ (seen as a cycle of $\mathsf{Compass}_{\mathfrak{R}}(W)$). Given $a \in \mathbb{N}$ and a graph G, let $\mathsf{ext}_a(G)$ denote the set of all pairs (G', A) such that $A \subseteq V(G')$ has size at most a and $G = G' - A$. We say that a flatness pair $(W, \mathfrak{R})$ of a graph G is *(a, ℓ)-homogeneous* if, for each $(G', A) \in \mathsf{ext}_a(G)$, $(W, \mathfrak{R})$, that is a flatness pair of $G = G' - A$, is ℓ-homogeneous with respect to A.

The following observation is a consequence of the fact that, given a wall W and a subwall W' of W, every internal brick of a tilt W'' of W' is also an internal brick of W.

Observation 4.6.9 ([285]). *Let $a, \ell \in \mathbb{N}$, G be a graph, and $(W, \mathfrak{R})$ be a flatness pair of G. If $(W, \mathfrak{R})$ is (a, ℓ)-homogeneous, then for every subwall W' of W, every W'-tilt of $(W, \mathfrak{R})$ is also (a, ℓ)-homogeneous.*

4.6.5 Tight renditions

A tight rendition is a vortex-free rendition of some society (G, Ω) with a few more properties, the main one being that there are $|\tilde{c}|$ disjoint paths from each cell c to $V(\Omega)$.

Tight renditions. We call *tight* rendition a vortex-free rendition ρ in the sphere such that the following conditions are satisfied:

1. if there are two points x, y of $N(\rho)$ such that $e = \{\pi_\rho(x), \pi_\rho(y)\} \in E(G)$, then there is a cell $c \in C(\rho)$ such that $\sigma(c)$ is the two-vertex connected graph $(e, \{e\})$,
2. for every $c \in C(\rho)$, every two vertices in $\pi_\rho(\tilde{c})$ belong to some path of $\sigma(c)$,
3. for every $c \in C(\rho)$ and every connected component C of the graph $\sigma(c) - \pi_\rho(\tilde{c})$, if $N_{\sigma_\rho(c)}(V(C)) \neq \emptyset$, then $N_{\sigma(c)}(V(C)) = \pi_\rho(\tilde{c})$,
4. there are no two distinct non-trivial cells c_1 and c_2 such that $\pi_\rho(\tilde{c}_1) = \pi_\rho(\tilde{c}_2)$, and
5. for every $c \in C(\rho)$, there are $|\tilde{c}|$ vertex-disjoint paths in G from $\pi_\rho(\tilde{c})$ to the set $V(\Omega)$.

We say that a flatness pair is *tight* if the underlying rendition is tight.

Because of the next result, renditions are often implicitly assumed to be tight in papers such as [24, 286].

Proposition 4.6.10 ([286])**.** *There is a linear-time algorithm that, given a vertex-free rendition of a society (G, Ω), outputs a tight rendition of (G, Ω).*

Irrelevant sets. Let G be a graph and let $\ell \in \mathbb{N}$. We say that a vertex set $X \subseteq V(G)$ is *ℓ-irrelevant* if every graph H with detail at most ℓ that is a minor of G is also a minor of $G - X$.

Linkages. A *linkage* L of *order* k in a graph G is the union of a collection of k pairwise vertex-disjoint paths of G. The set of pairs of vertices corresponding to the endpoints of these paths is the *pattern* of L. The Unique Linkage Theorem, proven in [279, 281] and also [196], asserts that there is a function f_{ul} such that if L is a linkage of pattern $\mathcal{P}$ of order k in a graph G with $V(G) = V(L)$ and L is unique with pattern $\mathcal{P}$, then the treewidth of G is at most $f_{\mathsf{ul}}(k)$. The linkage function appears in the general dependency of several results related to the application of the irrelevant vertex technique (see [4, 24, 120, 144, 146, 285, 286]).

We state the following result from [24]. In fact, [24, Theorem 5.9] is stated for boundaried graphs. Proposition 4.6.11 is derived by the same proof if we consider graphs with empty boundary.

Proposition 4.6.11 ([24])**.** *There exist two functions $f_{4.6.11} : \mathbb{N}^3 \to \mathbb{N}$ and $g_{4.6.11} : \mathbb{N}^2 \to \mathbb{N}$, where the images of $f_{4.6.11}$ are odd numbers, such that the following holds.*

Let $a, \ell \in \mathbb{N}$, $q \in \mathbb{N}_{\geq 3}$ be an odd integer, and G be a graph. Let A be a subset of $V(G)$ of size at most a and $(W, \mathfrak{R})$ be a regular tight flatness pair of $G - A$ of height at least $f_{4.6.11}(a, \ell, q)$ that is $g_{4.6.11}(a, \ell)$-homogeneous with respect to A.

Then the vertex set of the compass of every $W^{(q)}$-tilt of $(W, \mathfrak{R})$ is ℓ-irrelevant.

Moreover, it holds that $f_{4.6.11}(a, \ell, q) = \mathcal{O}((f_{\mathsf{ul}}(16a + 12\ell))^3 + q)$ and $g_{4.6.11}(a, \ell) = a + \ell + 3$, where f_{ul} is the function of the Unique Linkage Theorem.

The following result states that a tight and homogeneous flatness pair can be found inside any big enough flatness pair. Actually, the result was stated in [284] (and previously in [286]) for ℓ-homogeneity with respect to every subset of A, but the proofs all work the same way for the more general case of (a, ℓ)-homogeneity.

Proposition 4.6.12 ([284])**.** *There is a function $f_{4.6.12} : \mathbb{N}^4 \to \mathbb{N}$, whose images are odd integers, and an algorithm with the following specifications:*

$\texttt{Homogeneous}(r, a, \ell, t, G, W, \mathcal{R})$
***Input**: Integers $r \in \mathbb{N}_{\geq 3}$, $a, \ell, t \in \mathbb{N}$, a graph G, and a flatness pair $(W, \mathfrak{R})$ of G of height $f_{4.6.12}(r, a, \ell)$ whose $\mathfrak{R}$-compass has treewidth at most t.*
***Output**: A flatness pair $(\breve{W}, \breve{\mathfrak{R}})$ of G of height r that is tight, (a, ℓ)-homogeneous, and is a W'-tilt of $(W, \mathfrak{R})$ for some subwall W' of W.*
Moreover, $f_{4.6.12}(r, a, \ell) = \mathcal{O}(r^{g_{4.6.12}(a, \ell)})$ where $g_{4.6.12}(a, \ell) = 2^{2^{\mathcal{O}((a+\ell)\cdot\log(a+\ell))}}$ and the algorithm runs in time $2^{\mathcal{O}(g_{4.6.12}(a,\ell)\cdot r\log r + t\log t)} \cdot (n + m)$.

Part II

A structure theorem

CHAPTER 5

Excluding pinched spheres

Contents

In this chapter, we prove the results presented in Section 2.1, which are restated here for convenience.

Theorem 2.1.1. *Let H be an edge-apex graph. Then there is a constant c_H such that, if a graph G is H-minor-free, then $\mathsf{p}^\star(G) \leq c_H$, where p is the parameter mapping each graph F to the minimum k such that F is k-almost embeddable in the projective plane.*

Theorem 2.1.2. *Let H be an edge-apex graph. There is a constant c_H such that, if a graph G is H-minor-free, then $\mathsf{idpr}^\star(G) \leq c_H$.*

Theorem 2.1.3. *For any $h \in \mathbb{N}$, there is an edge-apex graph H such that $\mathsf{idpr}^\star(H) \geq h$.*

Theorem 2.1.4. *For every $H_1 \in \mathcal{G}_{\mathsf{projective}}$ and every $H_2 \in \mathcal{G}_{\mathsf{edge\text{-}apex}}$, there exist a constant c_{H_1,H_2} such that, if a graph G excludes both H_1 and H_2 as a minor, then $\mathsf{idpl}^\star(G) \leq c_{H_1,H_2}$.*

More particularly, in Section 5.1, we redefine the structure theorem using the terminology of "parametric graphs" and give an overview of the proof structure. In Section 5.2, we prove Theorem 2.1.1 and deduce from it Theorem 2.1.2 and Theorem 2.1.4. Finally, we prove Theorem 2.1.3 in Section 5.3.

5.1 Proof structure

An important tool in our proofs is to express the class $\mathcal{G}_{\mathsf{edge\text{-}apex}}$ using parametric graphs.

Parametric graphs. A *parametric graph* is a sequence $\mathscr{H} := \langle \mathscr{H}_t \rangle_{t\in\mathbb{N}}$ of graphs such that, for every $t \in \mathbb{N}$, $\mathscr{H}_t$ is a minor of $\mathscr{H}_{t+1}$. Given two parametric graphs $\mathscr{H}^1 = \langle \mathscr{H}^1_t \rangle_{t\in\mathbb{N}}$ and $\mathscr{H}^2 = \langle \mathscr{H}^2_t \rangle_{t\in\mathbb{N}}$, we write $\mathscr{H}^1 \lesssim \mathscr{H}^2$ if there is a function $f\colon \mathbb{N} \to \mathbb{N}$ such that, for every $k \in \mathbb{N}$, $\mathscr{H}^1_k$ is a minor of $\mathscr{H}^2_{f(k)}$. In case f is a linear function, we write $\mathscr{H}^1 \lesssim_{\mathsf{L}} \mathscr{H}^2$. We say that $\mathscr{H}^1$ and $\mathscr{H}^2$ are *equivalent* if $\mathscr{H}^1 \lesssim \mathscr{H}^2$ and $\mathscr{H}^2 \lesssim \mathscr{H}^1$. When we can replace $\lesssim$ by $\lesssim_{\mathsf{L}}$ then we say that $\mathscr{H}^1$ and $\mathscr{H}^2$ are *linearly equivalent.*

Given a parametric graph $\mathscr{H} := \langle \mathscr{H}_t \rangle_{t\in\mathbb{N}}$, we define the parameter $\mathsf{p}_{\mathscr{H}}\colon \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ so that $\mathsf{p}_{\mathscr{H}}(G)$ is the maximum k for which G contains $\mathscr{H}_k$ as a minor. Notice that $\mathscr{H}^1$ and $\mathscr{H}^2$ are equivalent if and only if $\mathsf{p}_{\mathscr{H}^1}$ and $\mathsf{p}_{\mathscr{H}^2}$ are equivalent. Moreover, notice that $\mathsf{p}_{\mathscr{H}}$ is a *minor-monotone* parameter.

Cylindrical grids and their enhancements. For every two non-negative integers t_1 and t_2, we define the $(t_1 \times t_2)$-*cylindrical grid* as the Cartesian product of a cycle on t_1 vertices and a path on t_2 vertices. Notice that the $(t_1 \times t_2)$-cylindrical grid contains t_2 cycles (each with t_1 vertices) and t_1 paths crossing them (each with t_2 vertices).

The *annulus grid* or order t, denoted by $\mathscr{A}_t$, is the $(4t \times t)$-cylindrical grid. We define four parametric graphs: The annulus grid $\mathscr{A} = \langle \mathscr{A}_t \rangle_{t\in\mathbb{N}}$, the single-cross grid $\mathscr{S} = \langle \mathscr{S}_t \rangle_{t\in\mathbb{N}}$, the long-jump grid $\mathscr{J} = \langle \mathscr{J}_t \rangle_{t\in\mathbb{N}}$, and the crosscap grid $\mathscr{C}_t$, where $\mathscr{S}_t$ (resp. $\mathscr{J}_t$, $\mathscr{C}_t$) is obtained if we add two (resp. $t+1$, $2t$) edges in $\mathscr{A}_t$, as indicated in Figure 5.1. In any case, we refer to the index t as the *order* of the corresponding parametric graph.

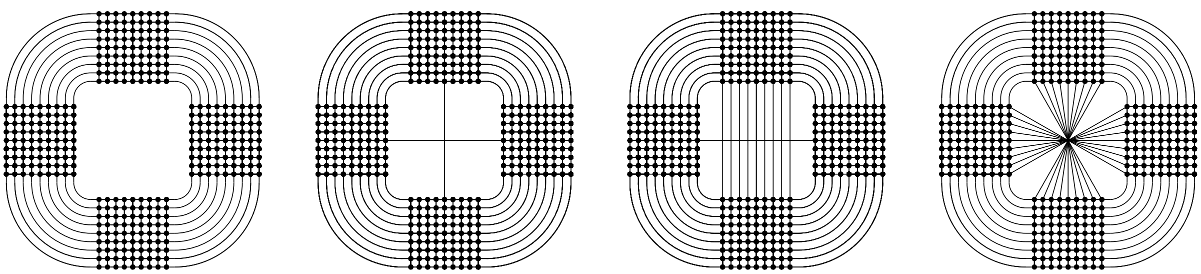

Figure 5.1: The annulus grid $\mathscr{A}_9$, the single-cross grid $\mathscr{S}_9$, the long-jump grid $\mathscr{J}_9$, and the crosscap grid $\mathscr{C}_9$.

As was already observed in [282, (1.5)], the class $\mathcal{G}_{\mathsf{planar}}$ of planar graphs are exactly the minors of the graphs in $\mathscr{A}$ (actually [282, (1.5)] is stated in terms of grids that, seen as parametric graphs, are equivalent to the annulus grids). With a simple adaptation of the proof of [282, (1.5)] (see also [136]), one may prove the following

Proposition 5.1.1. *$\mathcal{G}_{\mathsf{singly\text{-}crossing}}$ is the set of all minors of the graphs in $\mathscr{S}$ and $\mathcal{G}_{\mathsf{edge\text{-}apex}}$ is the set of all minors of the graphs in $\mathscr{J}$.*

Using the terminology of parametric graphs and Proposition 5.1.1, the answer to Question (1.2) for the case of $\mathcal{G}_{\mathsf{planar}}$ is size, because $\mathsf{p}_{\mathscr{A}} \sim \mathsf{size}^{\star}$ (from [261]), and the answer to Question (1.2) for the case of $\mathcal{G}_{\mathsf{singly\text{-}crossing}}$ is psize, because $\mathsf{p}_{\mathscr{S}} \sim \mathsf{psize}^{\star}$ (from [268]).

Our objective is to prove that the the answer to Question (1.2) for the case of $\mathcal{G}_{\mathsf{edge\text{-}apex}}$ is idpr, because $\mathsf{p}_{\mathscr{J}} \sim \mathsf{idpr}^{\star}$. Indeed, Theorem 2.1.2 is equivalent to $\mathsf{idpr}^{\star} \preceq \mathsf{p}_{\mathscr{J}}$ and Theorem 2.1.3 is equivalent to $\mathsf{p}_{\mathscr{J}} \preceq \mathsf{idpr}^{\star}$. The proof of the upper bound $\mathsf{idpr}^{\star} \preceq \mathsf{p}_{\mathscr{J}}$ is given in Section 5.2 and the proof of the lower bound $\mathsf{p}_{\mathscr{J}} \preceq \mathsf{idpr}^{\star}$ is given in Section 5.3.

For the proof of the upper bound $\mathsf{idpr}^{\star} \preceq \mathsf{p}_{\mathscr{J}}$, we assume that a graph G excludes $\mathscr{J}_k$ as a minor and our main task is to prove that G admits a tree decomposition where each torso is t-almost embeddable (see Subsection 5.2.3 for the formal definition) in the projective plane, for some t depending on k.

Our proof is inspired from the local structure theorem in [195], stating that for every “big enough” wall W in G there is a bounded-size vertex set A such that $G - A$ has a Σ-decomposition of bounded breadth and bounded depth, where Σ is a surface of bounded genus. We prove a local structural theorem, assuming the absence of $\mathscr{J}_k$ as a minor, that obtains a restriction of this Σ-decomposition so that Σ becomes the projective plane or the sphere. In Subsection 5.2.1 we give some preliminary results that permit to refine the structure of the graphs inside the vortices. These results are used in Subsection 5.2.2 in order to prove that the absence of $\mathscr{J}_k$ as a minor makes it possible to further modify the Σ-decomposition so that it becomes “apex-less”, i.e., $A = \emptyset$. This completes the proof of our local structure theorem. Our next aim is to transform our local structure theorem to a global one that is a tree decomposition where every torso with a “big enough” size is t-almost embeddable on the projective plane. It appears that the standard proof technique to go “from local to global” used in [87, 88, 101, 195, 303] does not work in our cases as it introduces new apices which, in our case, we need to avoid. For this, we propose a new approach of going from local to global without introducing apices. Our approach is presented in Subsection 5.2.3 and is tailor-made to the type of decomposition that we are looking for. It is based on the fact that the surface of our decomposition is the projective plane or the sphere and that these are the only two surfaces where the removal of a cycle creates a connected component that is a disk. The next step is to transform our global decomposition to one that bounds $\mathsf{idpr}(G)$. Let X be the set of all vertices in vortices. An important property of t-almost embeddablility, proved in [302], is that the bidimensionality of X in G_t is bounded by some function of t. As each vortex $G_t^i, i \in [t]$ is drawn in the closure of some face, we may identify all vertices of G_t that are drawn in the interior of this face to a single vertex and, by doing these identifications for all the vortices, obtain a projective graph. This completes the outline of the steps we used in order to prove that $\mathsf{idpr}^\star \preceq \mathsf{p}_{\mathscr{J}}$.

For the proof of the lower bound $\mathsf{p}_{\mathscr{J}} \preceq \mathsf{idpr}^\star$, we observe first that idpr, and therefore $\mathsf{idpr}^\star$ as well, is minor-monotone (Lemma 5.3.1), and we prove that $\mathsf{idpr}^\star(\mathscr{J}_k) = \Omega(k^\alpha)$ for some $\alpha > 0$ (Lemma 5.3.3). For the latter, in Subsection 5.3.1 using the main result of [82], we first prove that $\mathsf{idpr}(\mathscr{J}_k) = \Omega(k^{1/4})$ and then we use some general purpose result from [302] proving that this lower bound holds also for the clique-sum extension of idpr with a worse, however positive, exponent (Subsection 5.3.2).

We also stress that the proof of Theorem 2.1.4 follows an easy variant of all the steps above where the absence of a crosscap as a minor further imposes that the surfaces in the obtained decompositions are all spheres (see Theorem 5.2.27).

Convention on $\mathscr{J}$ and $\mathscr{C}$. Above, we defined the parametric graphs $\mathscr{J}$ (the long-jump grid) and $\mathscr{C}$ (the crosscap grid) presented in Figure 5.1. For ease of proof in the rest of the chapter, in place of $\mathscr{J}$ and $\mathscr{C}$, we will consider the parametric graphs given in Figure 5.2, that are obviously linearly equivalent to the ones presented above (they are more practical but are not as beautiful).

5.2 The upper bound

In this section, we prove Theorem 2.1.2. To prove this structure theorem, we proceed in three steps. We first prove a result (Theorem 5.2.20) on societies (the relevant definitions are provided in Section 4.5 and Section 5.2.1) in Subsection 5.2.1 that essentially says the following: a society (G, Ω) either contains a big long-jump transaction, or has a rendition in the projective plane with a bounded number of vortices, that are of small depth. We then combine our result on societies

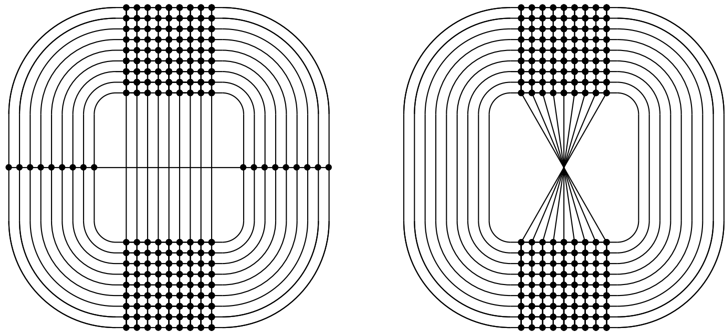

Figure 5.2: The long-jump grid $\mathscr{J}_k$ (left) and the crosscap grid $\mathscr{C}_k$ (right) for the rest of the chapter (here $k = 9$).

with a new version of the flat wall theorem (Theorem 5.2.25) to obtain a local structure theorem (Theorem 5.2.26) in Subsection 5.2.2 that does not have apex vertices and where the surface of the Σ-decomposition is the projective plane, i.e., in other words, it says the following: if G has big treewidth, then either G contains a big long-jump grid as a minor, or G has a Σ-decomposition with a bounded number of vortices, that are of small depth, where Σ is the projective plane. We then use in Subsection 5.2.3 our local structure theorem to prove a first global theorem (Theorem 2.1.1, and more precisely Theorem 5.2.32) that says that if G excludes a long-jump grid as a minor, then G has a tree decomposition such that the torso at each bag has an almost embedding in the projective plane with a bounded number of vortices, that are of small depth. Finally, we deduce from this first global structure theorem the one of Theorem 2.1.2 (more precisely Theorem 5.2.39) by identifying each vortex to a single vertex. As our local decomposition is "apex-less", this needs some special attention due to the fact that the classic "local to global" argument may introduce apices. In fact, we prove that a global decomposition without apices may be constructed. Interestingly, this is quite particular to the decomposition that we are looking for. It is essentially based to the fact that one of the connected components created after the removal of a cycle in the projective plane is always a disk.

Additionally, we also give similar results in case we exclude both a long-jump grid and a crosscap grid (Theorem 5.2.27, Theorem 5.2.37, Theorem 5.2.40). The results are similar (and simpler), as the surface is now the sphere instead of the projective plane.

5.2.1 Excluding a long-jump transaction from a society

The proof of the main result (Theorem 5.2.20) of this part is rather involved, so we provide here a quick sketch of the proof.

As said above, the goal here is to prove that a society (G, Ω) excluding a long-jump transaction (hence such that G excludes a long-jump grid has minor) has a rendition in the *projective plane* with a bounded number of vortices of small depth (Section 5.2.1, Theorem 5.2.20). We use many techniques from [195], where Kawarabayashi, Thomas, and Wollan in particular prove that a society (G, Ω) such that G excludes a big clique as a minor has a rendition in the some bounded genus surface, with a bounded number of apices, and a bounded number of vortices, all of small depth. Given that we exclude here a simpler graph, a lot of results in terms of cliques can be simplified by recognizing the long-jump under many forms (Section 5.2.1, Lemma 5.2.4). In particular, our first aim is to get rid of apices.

To prove Theorem 5.2.20, we first prove that a society (G, Ω) excluding a long-jump transaction either has a crosscap transaction $\mathcal{Q}$, or has a rendition in the *plane* with a bounded number of vortices, that are of small depth (Section 5.2.1, Theorem 5.2.16). In the second case, we can immediately conclude. In the first case, we find another society (G', Ω') inside (G, Ω) avoiding $\mathcal{Q}$, to which we again apply Section 5.2.1, Theorem 5.2.16. If we find a second crosscap transaction, then we can actually prove that (G, Ω) would contain a long-jump, a contradiction. Otherwise, (G', Ω') has a rendition in the *plane* with a bounded number of vortices of small depth, and thus (G, Ω) has a rendition in the *projective plane* with a bounded number of vortices of small depth.

To prove Theorem 5.2.16, we first observe that if (G, Ω) excludes a long-jump of order k as a minor, then it either has a rendition in the plane with a unique vortex of small depth, or contains a big crosscap transaction, or contains a big planar transaction $\mathcal{Q}$. In the first two cases, we can already conclude. In the third case, there is $\mathcal{Q}' \subseteq \mathcal{Q}$ such that the (*strip*) society $(G_{\mathcal{Q}'}, \Omega_{\mathcal{Q}'})$ corresponding to $\mathcal{Q}'$ (Section 5.2.1, Lemma 5.2.6) has a vortex-free rendition in the plane. In this case, we split (G, Ω) into two societies (G_1, Ω_1) and (G_2, Ω_2) so that they are separated by $(G_{\mathcal{Q}'}, \Omega_{\mathcal{Q}'})$ (Section 5.2.1, Lemma 5.2.13). For $i \in [2]$, if k_i is the maximum order of a long-jump in (G_i, Ω_i), then (G_i, Ω_i) excludes a long-jump of order $k_i + 1$. Given that (G, Ω) excludes a long-jump of order k, we can actually prove that $k_i + 1 < k$. We can thus recurse on (G_1, Ω_1) and (G_2, Ω_2), excluding a long-jump of order $k_1 + 1$ and $k_2 + 1$, respectively. If one of them contains a big crosscap transaction, then we can conclude. Otherwise, both have a rendition in the *plane* with a bounded number of vortices of small depth, where the number of vortices and the depth depends of $k_i + 1$. We can then combine both renditions along with the one of $(G_{\mathcal{Q}'}, \Omega_{\mathcal{Q}'})$ to get a rendition in the *plane* with a bounded number of vortices of small depth (depending on k).

Transactions in societies

Let us define transactions and cylindrical renditions and give additional results.

Linkages. Let G be a graph. A *linkage* in G is a set of pairwise vertex-disjoint paths. In slight abuse of notation, if $\mathcal{L}$ is a linkage, we use $V(\mathcal{L})$ and $E(\mathcal{L})$ to denote $\bigcup_{L \in \mathcal{L}} V(L)$ and $\bigcup_{L \in \mathcal{L}} E(L)$ respectively. Given two sets A and B, we say that a linkage $\mathcal{L}$ is an *A-B-linkage* if every path in $\mathcal{L}$ has one endpoint in A and one endpoint in B. We call $|\mathcal{L}|$ the *size* of $\mathcal{L}$.

Transactions. Let (G, Ω) be a society. A *transaction* in (G, Ω) is an A-B-linkage for disjoint segments A, B of Ω. We define the *depth* of (G, Ω) as the maximum order of a transaction in (G, Ω).

Let $\mathcal{T}$ be a transaction in a society (G, Ω). We say that $\mathcal{T}$ is *planar* if no two members of $\mathcal{T}$ form a cross in (G, Ω). An element $P \in \mathcal{T}$ is *peripheral* if there exists a segment X of Ω containing both endpoints of P and no endpoint of another path in $\mathcal{T}$. A transaction is *crooked* if it has no peripheral element.

Let $\mathcal{T} = \{P_1, ..., P_t\}$ be a transaction between segments A and B. For $i \in [t]$, let a_i (resp. b_i) be the endpoint of P_i in A (resp. B). Up to a permutation, we may assume that $a_1, ..., a_t$ occur in this order in Ω. We say that $\mathcal{T}$ is a *t-crosscap* transaction if $b_1, ..., b_t$ occur in this order in Ω. We say that $\mathcal{T}$ is a *crosscap* transaction if it is a t-crosscap transaction for any $t \in \mathbb{N}$. A transaction is called *monotone* if it is either a planar or a crosscap transaction. Suppose $t \geq 1$. We say that $\mathcal{T}$ is a *$(t-1)$-long-jump* transaction if $b_1, b_t, b_{t-1}, \ldots, b_3, b_2$ occur in this order in Ω. In other words, P_1 crosses the $t - 1$ paths of the planar transaction $\{P_2, ..., P_t\}$.

Proposition 5.2.1 ([107]). *Let $r, s \in \mathbb{N}$. Let (G, Ω) be a society. Let $\mathcal{Q}$ be a transaction in (G, Ω) of size $(r-1)(s-1)+1$. Then $\mathcal{Q}$ contains either a planar transaction $\mathcal{Q}' \subseteq \mathcal{Q}$ of size r, or a crosscap transaction $\mathcal{Q}' \subseteq \mathcal{Q}$ of size s.*

Vortex societies. Let Σ be a surface and G be a graph. Let $\delta = (\Gamma, \mathcal{D})$ be a Σ-decomposition of G. Every vortex c defines a society $(\sigma(c), \Omega)$, called the *vortex society* of c, by saying that Ω consists of the vertices in $\pi_\delta(\tilde{c})$ in the order given by Γ (there are two possible choices of Ω, namely Ω and its reversal. Either choice gives a valid vortex society). The *breadth* of δ is the number of cells $c \in C(\delta)$ which are a vortex and the *depth* of Δ is the maximum depth of the vortex societies $(\sigma(c), \Omega)$ over all vortex cells $c \in C(\Delta)$.

Cylindrical renditions. Let (G, Ω) be a society, $\rho = (\Gamma, \mathcal{D})$ be a rendition of (G, Ω) in a disk, and let $c_0 \in C(\rho)$ be such that no cell in $C(\rho) \setminus \{c_0\}$ is a vortex. We say that the triple $(\Gamma, \mathcal{D}, c_0)$ is a *cylindrical rendition* of (G, Ω) around c_0.

Proposition 5.2.2 (Lemma 3.6, [195])**.** *Let (G, Ω) be a society and $p \geq 4$ be a positive integer. Then (G, Ω) has a crooked transaction of size p, or a cylindrical rendition of depth at most $6p$.*

Nests and railed nests. Let $\delta = (\Gamma, \mathcal{D})$ be a Σ-decomposition of a graph G in a surface Σ and let $\Delta \subseteq \Sigma$ be an arcwise connected set. A *nest in δ around Δ of order s* is a sequence $\mathcal{C} = \langle C_1, C_2, \ldots, C_s \rangle$ of disjoint cycles in G such that each of them is grounded in δ and the track of C_i bounds a closed disk Δ_{C_i} in such a way that $\Delta \subseteq \Delta_{C_1} \subsetneq \Delta_{C_2} \subsetneq \cdots \subsetneq \Delta_{C_s} \subseteq \Sigma$. We call C_1 (resp. C_s) the *internal* (resp. *external*) cycle of $\mathcal{C}$. We call the sequence $\langle \Delta_{C_1}, \Delta_{C_2}, \ldots, \Delta_{C_s} \rangle$ the *disk sequence* of the nest in δ around Δ. If $\delta = (\Gamma, \mathcal{D}, c_0)$ is a cylindrical rendition, then we say that $\mathcal{C}$ is a nest in ρ around c_0.

Moreover, let $A = V(C_1) \cap \pi_\delta(N(\delta))$, $B = V(C_s) \cap \pi_\delta(N(\delta))$, and assume that $\{P_1, \ldots, P_r\}$ is an A-B-linkage such that for every $(i, j) \in [s] \times [r]$ the graph $C_i \cap P_j$ is a (possibly edgeless) path. We call the pair $(\mathcal{C}, \mathcal{P})$ a *railed nest in δ around Δ of order (s, r)*. Notice that $\bigcup \mathcal{P}$ is disjoint from $\mathsf{inner}_\delta(C_1) - V(C_1)$ and $\mathsf{outer}_\delta(C_s) - V(C_s)$.

Orthogonal and unexposed transactions. Let $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition in a society (G, Ω). Let $\mathcal{C}$ be a nest in ρ around c_0 and $\mathcal{Q}$ be a transaction in (G, Ω). We say that $\mathcal{Q}$ is *orthogonal* to $\mathcal{C}$ if, for each $C \in \mathcal{C}$ and each $Q \in \mathcal{Q}$, the graph $C \cap Q$ has at most two components. We say that $\mathcal{Q}$ is *unexposed* in ρ if each $Q \in \mathcal{Q}$ has at least one edge in $\sigma(c_0)$. Note that any crooked transaction is necessarily unexposed. If $\mathcal{Q}$ is orthogonal and unexposed, then every element of $\mathcal{Q}$ contains exactly two disjoint minimal subpaths which each have one endpoint in $V(\Omega)$ and the other in $V(\sigma(c_0))$. Let $\mathcal{P}$ be the union of all such minimal subpaths over the elements of $\mathcal{Q}$. $\mathcal{P}$ is called the *rail truncation* of $\mathcal{Q}$. Note that $(\mathcal{C}, \mathcal{P})$ is a railed nest.

Coterminal transactions. Let $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition in a society (G, Ω). Let $(\mathcal{C} = (C_1, \ldots, C_s), \mathcal{P})$ be a railed nest in ρ around c_0 and $\mathcal{Q}$ be a transaction in (G, Ω). We say that $\mathcal{Q}$ is *coterminal* with $\mathcal{P}$ up to level C_i if there exists a subset $\mathcal{P}' \subseteq \mathcal{P}$ such that $\mathsf{outer}_\rho(C_i) \cap \mathcal{Q} = \mathsf{outer}_\rho(C_i) \cap \mathcal{P}$. When it is clear from the context which nest we are referring to, we say that $\mathcal{Q}$ is *coterminal* with $\mathcal{P}$ up to level i.

Proposition 5.2.3 (Lemma 4.5, [195])**.** *Let r, s be positive integers with $s \geq 2r + 7$. Let (G, Ω) be a society and $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω). Let $((C_1, C_2, \ldots, C_s), \mathcal{P})$ be a railed nest of order $(s, 4r + 6)$ in ρ around c_0. If there exists a crooked transaction of size at least r in (G, Ω), then there exists a crooked transaction of size at least r in (G, Ω) that is coterminal with $\mathcal{P}$ up to level C_{2r+7}.*

Recognizing a long-jump

In this part, we define a few more parametric graphs (resp. types of transactions), and prove that they all contain a long-jump grid as a minor (resp. a long-jump transaction).

We consider here five new parametric graphs defined by adding edges in the *double-parameterized annulus grid* $\Gamma = \langle \Gamma_{k,r} \rangle$, where $\Gamma_{k,r}$ is the $(k \times r)$-cylindrical grid, as indicated in Figure 5.3. These

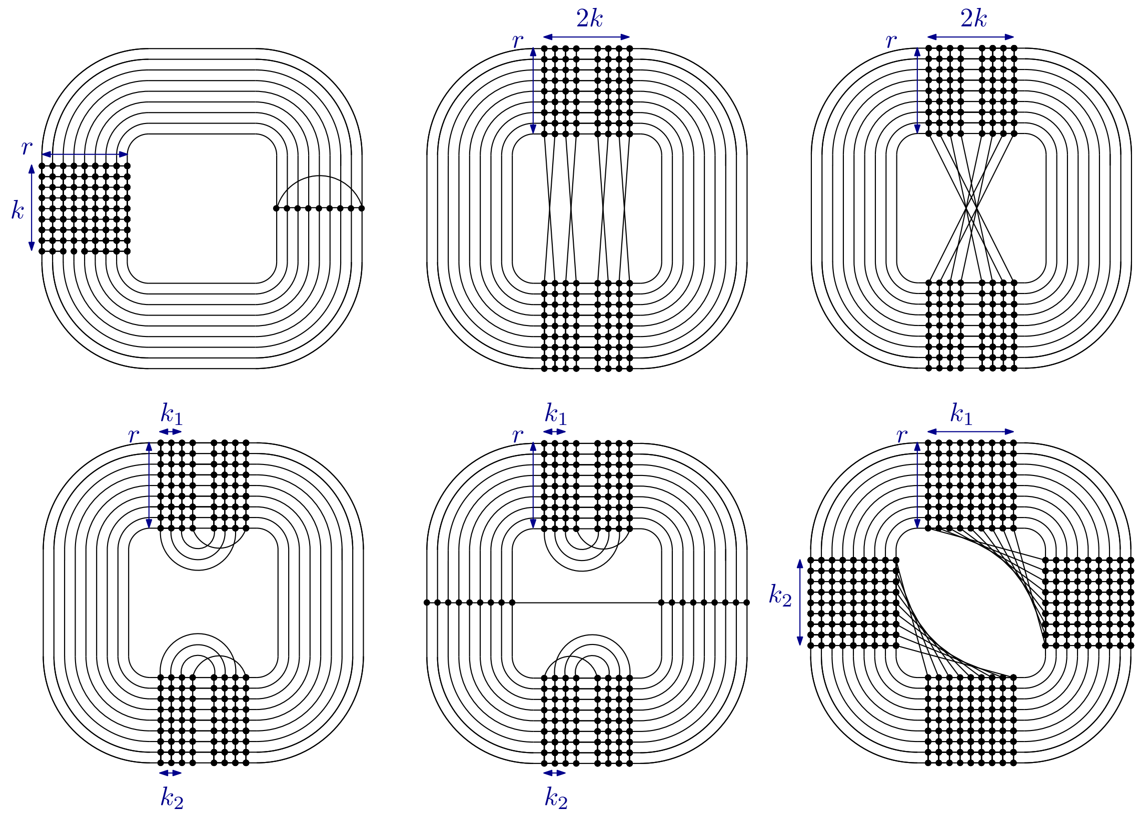

Figure 5.3: From up to down and left to right: the alternative jump $\hat{\mathscr{J}}^r_k$, the nested-crosses grid $\mathscr{NC}^r_k$, the twisted-crosses grid $\mathscr{TC}^r_k$, the double-jump grid $\mathscr{P}^r_{k_1,k_2}$, the alternative double-jump grid $\mathscr{Q}^r_{k_1,k_2}$, and the klein grid $\mathscr{K}^r_{k_1,k_2}$.

are the the alternative jump grid $\hat{\mathscr{J}}^r_k$,the nested-crosses grid $\mathscr{NC}^r_k$, the twisted-crosses grid $\mathscr{TC}^r_k$, the double-jump grid $\mathscr{P}^r_{k_1,k_2}$, the alternative double-jump grid $\mathscr{Q}^r_{k_1,k_2}$, and the klein grid $\mathscr{K}^r_{k_1,k_2}$.

We also define the transactions corresponding to the grids $\mathscr{NC}^r_k$, $\mathscr{TC}^r_k$, $\mathscr{P}^r_{k_1,k_2}$, $\mathscr{Q}^r_{k_1,k_2}$, and $\mathscr{K}^r_{k_1,k_2}$, respectively. Let (G, Ω) be a society and $k, k_1, k_2 \in \mathbb{N}$. Let A and B be two segments in (G, Ω) and $\mathcal{P} = \{P_1, \ldots, P_t\}$ be a transaction in (G, Ω) between A and B such that the endpoints of P_i are a_i and b_i with $a_1, \ldots, a_t$ occurring in A in this order.

- $\mathcal{P}$ is a *k-nested crosses transaction* if $t = 2k$ and for all $1 \leq i < j \leq 2k$, P_i and P_j cross if and only if i is odd and $j = i + 1$.

- $\mathcal{P}$ is a *twisted k-nested crosses transaction* if $t = 2k$ and for all $1 \leq i < j \leq 2k$, P_i and P_j do not cross if and only if i is odd and $j = i+1$. In other words, $b_2, b_1, b_4, b_3, \ldots, b_{2k-2}, b_{2k-3}, b_{2k}, b_{2k-1}$ occur in Ω in this order.

- $\mathcal{P}$ is a (k_1, k_2)-*double-jump transaction* if $t = k_1 + k_2 + 2$ and $\mathcal{P}$ can be partitioned into two planar transactions $\mathcal{Q}_1$ and $\mathcal{Q}_2$ of size k_1 and k_2 respectively and two isolated paths Q_1 and Q_2 such that $\mathcal{Q}'_i := \mathcal{Q}_i \cup Q_i$ is a k_i-long-jump transaction for $i \in [2]$, $\mathcal{Q}'_1$ and $\mathcal{Q}'_2$ do not cross, and there is an endpoint q_1 of Q_1 and an endpoint q_2 of Q_2 such that $\mathcal{P} \setminus \{Q_1, Q_2\}$ is a either a $q_1\Omega q_2$-linkage or a $q_2\Omega q_1$-linkage. In other words, no path of $\mathcal{P} \setminus \{Q_1, Q_2\}$ has an endpoint in one of $q_1\Omega q_2$ and $q_2\Omega q_1$.

- $\mathcal{P}$ is an *alternative* (k_1, k_2)-*double-jump transaction* if $t = k_1 + k_2 + 3$ and $\mathcal{P}$ can be partitioned into two planar transactions $\mathcal{Q}_1$ and $\mathcal{Q}_2$ of size k_1 and k_2 respectively and three isolated paths Q, Q_1, and Q_2 such that $\mathcal{Q}'_i := \mathcal{Q}_i \cup Q_i$ is a k_i-long-jump transaction for $i \in [2]$, for any endpoint q_1 of Q_1 and endpoint q_2 of Q_2, $\mathcal{P} \setminus \{Q_1, Q_2\}$ is a neither a $q_1\Omega q_2$-linkage nor a $q_2\Omega q_1$-linkage, and if a and b are the endpoints of Q, then one of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ is a $a\Omega b$-linkage, and the other is a $b\Omega a$-linkage.

- Finally, $\mathcal{P}$ is an (k_1, k_2)-*klein transaction* if $t = k_1 + k_2$ and $\mathcal{P}$ can be partitioned into two crosscap transactions $\mathcal{Q}_1$ and $\mathcal{Q}_2$ of size k_1 and k_2 respectively that do not cross.

Lemma 5.2.4. *Let $k, k_1, k_2, r \in \mathbb{N}$ such that $k_1 + k_2 \geq k$. Then $\mathscr{J}_k^r$ is a minor of $\hat{\mathscr{J}}_{2r}^{2r+k-2}$, $\mathscr{N}\mathscr{C}_k^{r+1}$, $\mathscr{T}\mathscr{C}_{k+2r}^{r+1}$, $\mathscr{P}_{k_1,k_2}^{r+1}$, $\mathscr{Q}_{k_1,k_2}^{r+1}$, and $\mathscr{K}_{k,k+1}^{r+k}$.*

Proof. For this proof, an illustration is more convenient than a long description. See Figure 5.4. In each grid, we need to find a path (in orange in Figure 5.4), jumping over a parallel linkage (in blue). The annulus grid is represented in green. $\square$

Lemma 5.2.5. *Let $k, k_1, k_2, r \in \mathbb{N}$ such that $k_1 + k_2 \geq k$. Let (G, Ω) be a society and $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω) with a railed nest $(\mathcal{C}, \mathcal{P})$ in ρ around c_0, where $\mathcal{C} = (C_1, \ldots, C_s)$ is of order $s \geq a$ and $a = 2$ for items (i)-(iv) and $a = k + 1$ for item (v). Let $\mathcal{Q}$ be a transaction in (G, Ω) orthogonal to $\mathcal{P}$ that is either:*

(i) a k-nested crosses transaction, or

(ii) a twisted $3k$-nested transaction, or

(iii) a (k_1, k_2)-double-jump transaction, or

(iv) an alternative (k_1, k_2)-double-jump transaction, or

(v) a $(k, k + 1)$-klein transaction.

Then there is a k-long-jump transaction that is coterminal with $\mathcal{P}$ up to level a.

Proof. Let H be the graph obtained from the union of $\mathcal{C}$, $\mathcal{P}$, $\mathcal{Q}$, and a cycle C with vertex set $V(\Omega)$ and an edge uv between any two consecutive vertices u, v in the cyclic ordering of Ω (note that these edges may not be edges in G). If $\mathcal{Q}$ is one of items (i) to (v), then H is one of $\mathscr{N}\mathscr{C}_k^{s+1}$, $\mathscr{T}\mathscr{C}_k^{s+1}$, $\mathscr{P}_{k_1,k_2}^{s+1}$, $\mathscr{Q}_{k_1,k_2}^{s+1}$, and $\mathscr{K}_{k_1,k_2}^{s+1}$. Then, by Lemma 5.2.4, $\mathscr{J}_k^{s-a+2}$ is a minor of H. Let X be the set of vertices in the cycles C and $C_a, \ldots, C_s$. More particularly, if we look at Figure 5.4, $\mathscr{J}_k^{s-a+2}$ is an X-minor of H. Hence, we deduce that there is $\mathcal{Q}' \subseteq \mathcal{Q}$ that is a k-long-jump transaction that is coterminal with $\mathcal{P}$ up to level C_a. $\square$

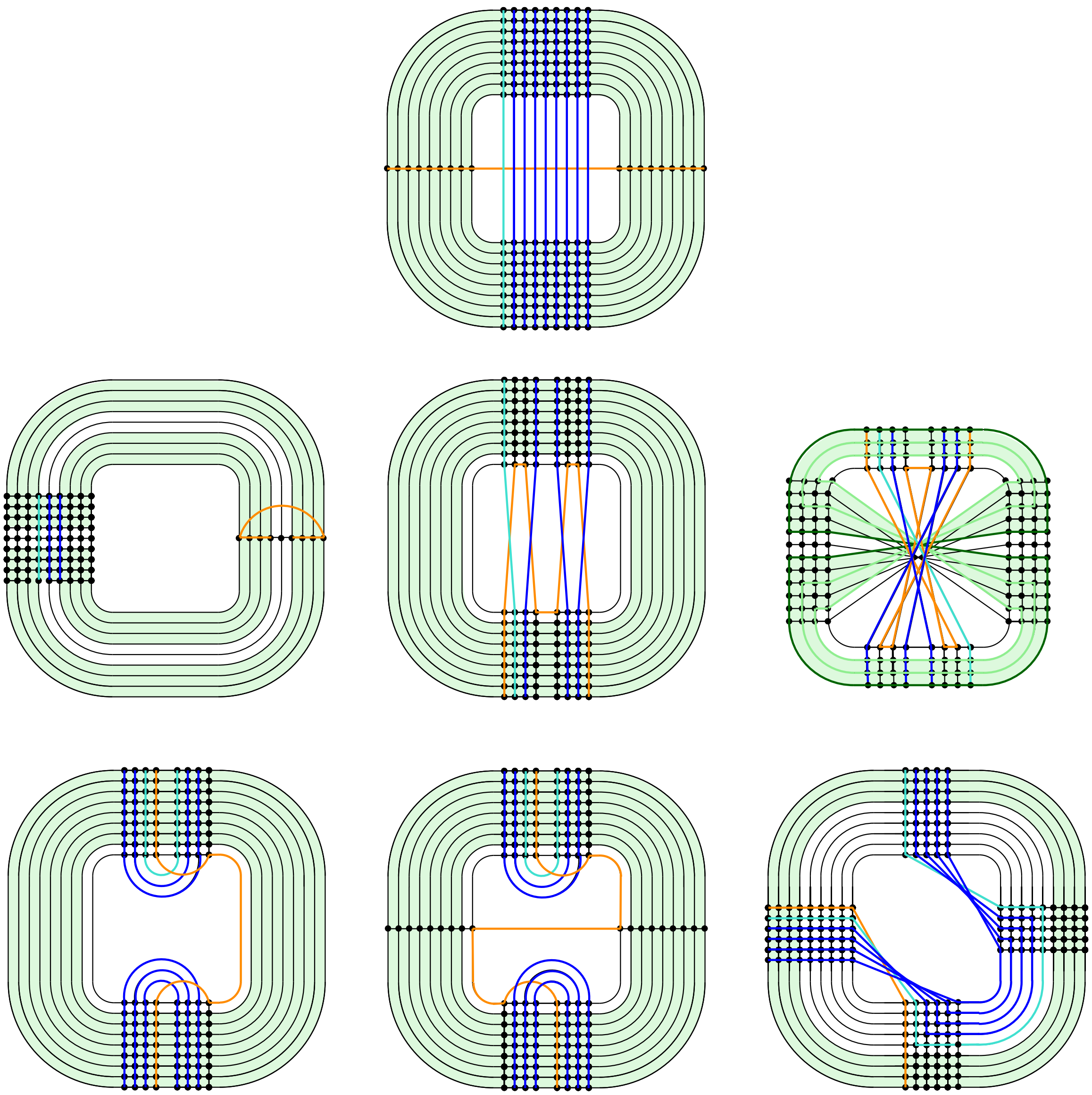

Figure 5.4: Proof of Lemma 5.2.4.

Finding an isolated and rural strip in a vortex

In this part, we prove that a society (G, Ω) containing a planar transaction $\mathcal{Q}$ either contains a long-jump transaction, or is such that the society (called *strip society*) bordered by some strip $\mathcal{Q}' \subseteq \mathcal{Q}$ has a vortex-free rendition.

We first define H-bridges and strip societies (from [195]).

Bridges. Let H be a subgraph of a graph G. An *H-bridge* in G is a connected subgraph B of G such that none of its edges is an edge of H and either $E(B)$ consists of a unique edge with both endpoints in H, or for some connected component C in $G - V(H)$, $E(B)$ consists of all edges of G with at least one endpoint in $V(C)$. The vertices in $V(B) \cap V(H)$ are called the *attachments* of B.

Strip societies. Let $\mathcal{P} = (P_1, ..., P_m)$ be a monotone transaction of size $m \geq 2$ in a society (G, Ω). Let X_1 and X_2 be disjoint segments of Ω such that $\mathcal{P}$ is a linkage from X_1 to X_2. For each $i \in [m]$, the endpoints of P_i are denoted by a_i and b_i in such a way that $a_i \in X_1$, $b_i \in X_2$, and the endpoints occur in Ω in the order $a_1, a_2, \ldots, a_m, b_m, b_{m-1}, \ldots, b_1$ if $\mathcal{P}$ is a planar transaction, and in the order $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m$ if $\mathcal{P}$ is a crosscap transaction.

Let H denote the subgraph of G obtained from the union of the elements of $\mathcal{P}$ by adding the elements of $V(\Omega)$ as isolated vertices. Let H' be the subgraph of H consisting of $\mathcal{P}$, $a_1\Omega a_m$, and $b_1\Omega b_m$. Let us consider all H-bridges of G with at least one attachment in $V(H') \setminus V(P_1 \cup P_n)$, and for each such H-bridge B, let B' denote the graph obtained from B by deleting all attachments that do not belong to $V(H')$. Finally, let G_1 denote the union of H' and all the graphs B' as above.

If $\mathcal{P}$ is a planar transaction, let the cyclic permutation Ω_1 be defined by saying that $V(\Omega_1) = a_1\Omega a_m \cup b_m\Omega b_1$ and that the order on Ω_1 is the one induced by Ω. If $\mathcal{P}$ is a crosscap transaction, let the cyclic permutation Ω_1 be defined by saying that $V(\Omega_1) = a_1\Omega a_m \cup b_1\Omega b_m$ and that the order on Ω_1 is obtained by following $a_1\Omega a_m$ in the order given by Ω, and then following $b_1\Omega b_m$ in the *reverse order* from the one given by Ω.

Thus (G_1, Ω_1) is a society, and we call it the *$\mathcal{P}$-strip society of* (G, Ω) with respect to (X_1, X_2). When there can be no confusion as to the choice of the segments (X_1, X_2), we will omit specifying them.

We say that P_1 and P_m are the *boundary paths* of the $\mathcal{P}$-strip society (G_1, Ω_1). We say that the $\mathcal{P}$-strip society of (G, Ω) is *isolated* in G if no edge of G has one endpoint in $V(G_1) \setminus V(P_1 \cup P_m)$ and the other endpoint in $V(G) \setminus V(G_1)$. Thus (G_1, Ω_1) is isolated if and only if every H-bridge of G with at least one attachment in $V(H') \setminus V(P_1 \cup P_m)$ has all its attachments in $V(H')$.

The following lemma, that is very much inspired from [195, Theorem 5.11], says that if a society has a big monotone transaction $\mathcal{Q}$, then either it contains a long-jump transaction, or the $\mathcal{Q}'$-strip society defined by some $\mathcal{Q}' \subseteq \mathcal{Q}$ is isolated and rural.

Lemma 5.2.6. *Let $l, k, s \in \mathbb{N}$ with $s \geq 9$. Let (G, Ω) be a society. Let $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω). Let $\mathcal{Q}$ be an unexposed planar (resp. crosscap) transaction in (G, Ω) of order $l' \geq k(l + 3k)$ (resp. $l' \geq 3k(l + 3k)$). Let $(\mathcal{C} = (C_1, \ldots, C_s), \mathcal{P})$ be a railed nest of order $(s, 2l')$ in ρ around c_0 where $\mathcal{P}$ is the rail truncation of $\mathcal{Q}$. Let X_1, X_2 be disjoint segments of Ω such that $\mathcal{Q}$ is a linkage from X_1 to X_2. Then one of the following holds.*

(i) there is a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level C_9, or

(ii) there is a transaction $\mathcal{Q}' \subseteq \mathcal{Q}$ of size at least l such that the $\mathcal{Q}'$-strip society of (G, Ω) with respect to (X_1, X_2) is isolated and rural.

Proof. Assume the claim is false. Let $k' = k$ if $\mathcal{Q}$ is a planar transaction, and $k' = 3k$ if $\mathcal{Q}$ is a crosscap transaction. Let $X^1 \subseteq X_1$ and $X^2 \subseteq X_2$ be the two disjoint minimal segments of Ω such that every path in $\mathcal{Q}$ has one endpoint in X^1 and the other point in X^2. Let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_{l'}\}$, where the paths are numbered in such a way that their endpoints appear in X_1 (and therefore also in X_2) in order.

Let $I'_1, ..., I'_{k'}$ be intervals of length $l + 3k$ with union $[l']$. For $i \in [k']$, let I_i be obtained from I'_i by deleting the last k elements. Thus $|I_i| = l + 2k$.

For $i \in [k']$, let $\mathcal{Q}_i$ be the set $\{Q_j \mid j \in I_i\}$. For $i \in [k']$, and $j \in [2]$, let X_i^j be the minimum subset of X^j forming a segment of Ω containing an endpoint of every element of $\mathcal{Q}_i$. Let (H_i, Ω_i) be the $\mathcal{Q}_i$-strip society of (G, Ω) with respect to (X_i^1, X_i^2).

Let $i \in [k']$. We define a society (H'_i, Ω_i) which is closely related to the strip society (H_i, Ω_i) as follows. Let J be the graph consisting of the union of $V(\Omega)$ treated as isolated vertices and $\mathcal{Q}$. For every J-bridge B in G, let B' be the subgraph obtained by deleting all attachments of B not in $V(\mathcal{Q}_i) \cup X_i^1 \cup X_i^2$. Let $\alpha+1$ be the smallest value in I_i. Let H_i' be the union of $J[V(\mathcal{Q}_i) \cup X_i^1 \cup X_i^2]$ and B' for every J-bridge B in G with at least one attachment in $(V(\mathcal{Q}_i) \cup X_i^1 \cup X_i^2) \setminus V(Q_{\alpha+1} \cup Q_{\alpha+|I_i|})$ (that is, an attachment not in the first or last element of $\mathcal{Q}_i$). Note that the difference between H_i and H_i' is that, to define H_i, we consider $Q_i \cup V(\Omega)$-bridges, while for H_i', we consider $Q \cup V(\Omega)$-bridges.

Claim 5.2.7. *The subgraphs H_i' are pairwise vertex disjoint.*

Proof of claim. Assume the claim is false and let us show that there is a long-jump transaction of order k in (G, Ω). Let $i < i'$ be indices such that H_i' intersects $H_{i'}'$. Thus, there exists a path R in G with one endpoint x_i in $X_i^1 \cup X_i^2 \cup V(\mathcal{Q}_i)$, the other endpoint $x_{i'}$ in $X_{i'}^1 \cup X_{i'}^2 \cup V(\mathcal{Q}_{i'})$ and no internal vertex in $V(\Omega) \cup V(\mathcal{Q})$.

Let F be the subgraph formed by the union of C_1 and the inner graph of C_1. We fix the path R to minimize the number of edges not contained in $E(F)$. Let $\bar{R} = R \cap F$.

Consider a maximal subpath T of R with all internal vertices contained in $V(R) \setminus V(\bar{R})$. There are two possible cases given the cylindrical rendition and the fact that R is internally disjoint from $\mathcal{Q}$: either T has one endpoint in $\{x_i, x_{i'}\}$ and one endpoint in C_1 or, alternatively, T has both endpoints in a component of $C_1 - V(\mathcal{Q})$. By replacing any such subpath T with an appropriately chosen subpath of C_1, it follows that there exists a path R' contained in $C_1 \cup \bar{R}$ with endpoints x_i' and $x_{i'}'$ such that

- R' is internally disjoint from $\mathcal{Q}$ and,
- x_i' is contained in $V(\mathcal{Q}_i)$ and $x_{i'}'$ is contained in $V(\mathcal{Q}_{i'}')$.

Note that if x_i is in an element of $\mathcal{Q}$, we can choose x_i' to be in the same element of $\mathcal{Q}$ as x_i. If $x_i \in V(\Omega)$, we can choose x_i' to be in either of the two elements of $\mathcal{Q}$ closest to x_i on Ω.

We conclude, by the choice of R', that R' is a subgraph of F. Thus, both endpoints x_i' and $x_{i'}'$ are contained in $V(\mathcal{Q}) \cap V(F)$. This path R' along with the k paths $(Q_{\alpha+|I_i|+1}, ..., Q_{\alpha+|I_i|+k})$ as well as $\mathcal{Q}_i$, and $\mathcal{Q}_{i'}$, create a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level C_2, and hence C_9, which is outcome (i). See Figure 5.6 for an illustration of the proof of Claim 5.2.23, which is similar. ◇

Claim 5.2.8. *There exists an index $i \in [k']$, such that (H_i', Ω_i') is rural.*

Proof of claim. The proof is exactly the same as the one of [195, Theorem 5.11, Claim 3] with the specificity that for us, $Z = \emptyset$, and thus $\alpha = 0$ and $a = 5$. The idea is that if none of the (H_i', Ω_i') is rural, then each contain a cross. Thus, we will find a transaction $\mathcal{Q}'$ from X_1 to X_2 that is coterminal with $\mathcal{Q}$ up to level C_8, such that $\mathcal{Q}'$ is either a k'-nested crosses transaction (if $\mathcal{Q}$ is planar) or a twisted k'-nested crosses transaction (if $\mathcal{Q}$ is a crosscap transaction). Then, by Lemma 5.2.5, there is a k-long-jump transaction that is coterminal with $\mathcal{P}$ up to level C_9, which is outcome (ii) of the statement. ◇

We fix, for the remainder of the proof, an index i such that (H_i', Ω_i') is rural, and fix a vortex-free rendition ρ' of (H_i', Ω_i') in the disk Δ. Let $I^\star$ be the interval obtained from I_i by deleting the first and last k elements. Note that $|I^\star| = l$. Let the first element in $I^\star$ be $\beta + 1$. Let $\mathcal{Q}^\star$ be the set $\{Q_j \mid j \in I^\star\}$, and X^{1*} and X^{2*} be the minimal segments contained in X_i^1 and X_i^2, respectively, containing an endpoint of each element of $\mathcal{Q}^\star$. Let T_1 and T_l be the track of $Q_{\beta+1}$ and $Q_{\beta+l}$ in the rendition ρ'.

Of the three connected components of $\Delta \setminus (T_1 \cup T_l)$, let $\Delta^\star$ be the (unique) component whose boundary contains T_1 and T_l. Let $H^\star = \bigcup_{c \in C(\rho'):c \subseteq \Delta^\star} \sigma(c) \cup Q_{\beta+1} \cup Q_{\beta+l}$. Let $\Omega^\star$ be the cyclically ordered set with $V(\Omega^\star) = X^{1*} \cup X^{2*}$ obtained by restricting Ω'_i to $V(\Omega^\star)$.

The rendition ρ' restricted to the disk $\Delta^\star$ can be extended to a vortex-free rendition of $(H^\star, \Omega^\star)$ by mapping the vertices of $Q_{\beta+1}$ and $Q_{\beta+l}$ to the boundary, and thus the following claim immediately follows.

Claim 5.2.9. *$(H^\star, \Omega^\star)$ is rural.*

We now see that no edge to $V(H^\star)$ avoids the paths $Q_{\beta+1}$ and $Q_{\beta+l}$. Recall that the definition of the subgraph J of G is the subgraph consisting of the union of $V(\Omega)$ treated as isolated vertices and $\mathcal{Q}$.

Claim 5.2.10. *There does not exist an edge xy of G with x in $V(H^\star) \setminus (V(Q_{\beta+1}) \cup V(Q_l))$ and y in $V(G) \setminus V(H^\star)$.*

Proof of claim. Assume that such an edge xy exists. Given the rendition ρ', it follows that the edge $xy \notin E(H'_i)$, and thus the vertex y is the attachment of a J-bridge B in G such that $y \in (V(\Omega) \setminus V(H'_i)) \cup \bigcup_{j \notin I_i} V(Q_j)$ and B has an attachment x' in $V(\Omega^\star) \cup V(\mathcal{Q}^\star)$. Fix a path P in B from x' to y which is internally disjoint from $V(\Omega) \cup V(\mathcal{Q})$. Then, using P, $\mathcal{Q}^\star$, and $\{Q_j \mid j \in I_i \setminus I^\star\}$, we conclude that there is a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level C_2, a contradiction. $\diamond$

Finally, we have the following.

Claim 5.2.11. *The $\mathcal{Q}^\star$-strip society of (G, Ω) is a subgraph of $H^\star$.*

Proof of claim. The proof is exactly the same as the one of [195, Theorem 5.11, Claim 6] with $Z = \emptyset$. Essentially, if that happened, then it would contradict Claim 5.2.10. $\diamond$

The theorem now follows as the vortex-free rendition of $(H^\star, \Omega^\star)$ restricted to the $\mathcal{Q}^\star$-strip society shows that the strip society is rural, and Claim 5.2.10 shows that the $\mathcal{Q}^\star$-strip society is isolated. $\square$

Splitting a vortex

In this part, we prove that, if a society (G, Ω) contains a unique vortex c_0 and a strip society that is rural and isolated, then c_0 can be split into two vortices c_1 and c_2 of smaller size that are separated by the strip society.

We require the following technical result.

Proposition 5.2.12 (Lemma 5.15, [195])**.** *Let (G, Ω) be a society with a cylindrical rendition $\rho = (\Gamma, \mathcal{D}, c_0)$. Let $\mathcal{C} = (C_1, ..., C_s)$ be a nest in ρ and let $\mathcal{Q}$ be an unexposed monotone transaction in (G, Ω) of order at least 3 orthogonal to $\mathcal{C}$. Let (H, Ω_H) be the $\mathcal{Q}$-strip society in (G, Ω) and let Y_1 and Y_2 be the two segments of $\Omega \setminus V(H)$. Assume that there exists a linkage $\mathcal{P} = \{P_1, P_2\}$ such that P_i links Y_i and $V(\sigma(c_0))$ for $i \in \{1, 2\}$, $\mathcal{P}$ is disjoint from $\mathcal{Q}$, and $\mathcal{P}$ is orthogonal to $\mathcal{C}$.*

Let $i \geq 7$. Let (G', Ω') be the inner society of C_i and let $\mathcal{Q}'$ be the restriction of $\mathcal{Q}$ to (G', Ω'). Let $\rho' = (\Gamma', \mathcal{D}', c_0)$ be the restriction of ρ to be a cylindrical rendition of (G', Ω'). Then $\mathcal{Q}'$ is unexposed and monotone in ρ'. Moreover, $\mathcal{Q}'$ is a crosscap transaction if and only if $\mathcal{Q}$ is a crosscap transaction. If the $\mathcal{Q}$-strip society is rural and isolated in (G, Ω), then the $\mathcal{Q}'$-strip society is rural and isolated in (G', Ω').

The main result of this section is that, given a cylindrical rendition ρ of a society (G, Ω) and a planar transaction $\mathcal{Q}$ such that the $\mathcal{Q}$-strip society in (G, Ω) is rural and isolated, there is another rendition ρ' of (G, Ω) of breadth two containing two smaller cylindrical renditions, and otherwise vortex-free.

Lemma 5.2.13. *Let $k, l, m, s, s' \in \mathbb{N}$ such that $l \geq 1$, $m \geq \max\{2s'+5, k+2\}$, and $s \geq s'+8$. Let (G, Ω) be a society and $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω) in a disk Δ. Let X_1, X_2 be two disjoint segments of Ω and let Y_1, Y_2 be the two segments of $\Omega \setminus (X_1 \cup X_2)$. Let $\mathcal{Q}$ be an unexposed planar transaction from X_1 to X_2 of size m such that the $\mathcal{Q}$-strip society in (G, Ω) is isolated and rural. Let $(\mathcal{C} = (C_1, C_2, \dots, C_s), \mathcal{P} \cup \mathcal{P}_1 \cup \mathcal{P}_2)$ be a railed nest in ρ of order $(s, 2m+2l)$ around c_0 such that $\mathcal{P}$ is the rail truncation of $\mathcal{Q}$ and, for $i \in [2]$, $\mathcal{P}_i$ is a linkage in ρ from Y_i to $V(\sigma(c_0))$ of size l.*

Then either there is a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P} \cup \mathcal{P}_1 \cup \mathcal{P}_2$ up to level C_8, or there is a rendition ρ' of (G, Ω) in Δ with exactly two vortices c_1 and c_2, and two disjoint ρ'-aligned disks $\Delta'_1, \Delta'_2 \subseteq \Delta$ such that

- *for $i \in [2]$, $\rho_i = \rho'[\Delta'_i]$ is a cylindrical rendition of $(\mathsf{inner}_{\rho'}(\Delta'_i), \Omega_{\Delta'_i})$ with vortex c_i,*
- *there is a railed nest $(\mathcal{C}_i = (C^i_1, \dots, C^i_{s'}), \mathcal{P}'_i)$ of order (s', l) around c_i in ρ_i, where, for each $P \in \mathcal{P}'_i$, P is a subpath of an element of $\mathcal{P}_i$ and, for each $j \in [s']$, $P \cap C^i_j = P \cap C_{j+7}$,*
- *$(C_8, \dots, C_s)$ is a nest around both the closure of c_1 and the closure of c_2, and*
- *there is a path $Q \in \mathcal{Q}$ such that Δ'_1 and Δ'_2 are contained in different connected components of $\Delta \setminus T$, where T is the track of Q in ρ'.*

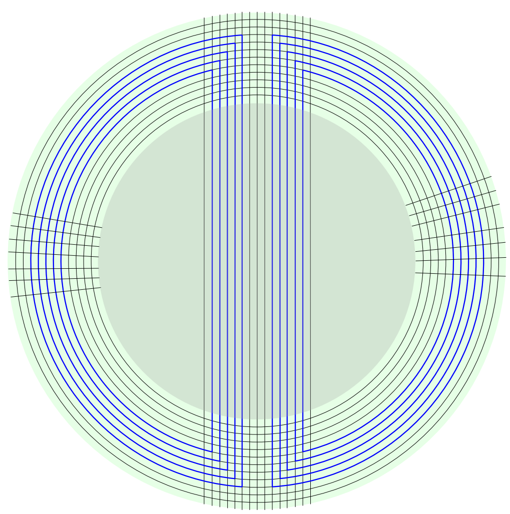

Figure 5.5: The rails, the nest, and the planar transaction of the original cylindrical rendition are represented in black and the rails and nests of the two new cylindrical renditions are represented in blue.

Proof. See Figure 5.5 for an illustration. Let us assume that G does not contain a long-jump transaction of order k in (G, Ω) that is coterminal with $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}$ up to level C_8. Let the elements of $\mathcal{Q}$ be enumerated $Q_1, Q_2, \dots, Q_m$ by the order in which their endpoints occur on X_1.

Restricting the strip society. We have $s \geq 7$. Let (H, Ω_H) be the inner society of C_7 in (G, Ω) with respect to the rendition ρ. Let Δ_H be the closed subdisk bounded by the track of C_7. Let $\mathcal{Q}_H$ be the restriction of $\mathcal{Q}$ to (H, Ω_H). Let (J, Ω_J) be the $\mathcal{Q}_H$-strip society in (H, Ω_H).

Given that $l \geq 1$, there are $P_1 \in \mathcal{P}_1$ and $P_2 \in \mathcal{P}_2$. By Proposition 5.2.12 applied to the society (G, Ω), linkages $\mathcal{Q}$ and $\{P_1, P_2\}$, and cycle C_7, the society (J, Ω_J) is rural and isolated in (H, Ω_H). There exists a vortex-free rendition ρ_J of (J, Ω_J) in Δ_H such that $\pi_\rho^{-1}(v) = \pi_{\rho_J}^{-1}(v)$ for all $v \in V(\Omega_J)$. Note that we are using the fact that $\mathcal{Q}$ is planar to ensure that the cyclic ordering Ω_J is the same as the cyclic ordering of $V(\Omega_J)$ in Ω_H.

Let T_1 (resp. T_2) be the track of Q_1 (resp. Q_m) restricted to (H, Ω_H) in ρ_J. There is a unique connected component of $\Delta_H \setminus (T_1 \cup T_2)$ whose boundary includes both T_1 and T_2; let Δ^* be the closure of this connected component. Let J' be the subgraph formed by $\bigcup\{\sigma_{\rho_J}(c) : c \in C(\rho_J), c \subseteq \Delta^*\}$ along with any vertices v of J such that $\pi_{\rho_J}^{-1}(v)$ exists and belongs to Δ^*. Let $\Omega_{J'}$ be the cyclically ordered set of vertices with $V(\Omega_{J'}) = V(\Omega_J) \cap V(J')$ with the cyclic order induced by Ω_J. Let $\rho_{J'}$ be the restriction of ρ_J to the disk Δ^*.

Defining a new vortex-free rendition $\rho^\star$ avoiding the two future vortices. We define the society (G', Ω) as follows. Let G' be the union of J' and the outer graph of C_7 with respect to ρ. Thus, the union of $\rho_{J'}$ along with the restriction of ρ to the complement of the interior of Δ_H gives us a vortex-free rendition $\rho^\star$ of (G', Ω) in the disk Δ.

Defining Δ_1 and Δ_2. Consider $\Delta_H \setminus \Delta^*$. There is one connected component which contains T_1 in its boundary and one which contains T_2 in its boundary. For $i \in [2]$, let U_i be the set of vertices $u \in V(G')$ such that $\pi_{\rho*}^{-1}(u)$ exists and is contained in the boundary of the connected component of $\Delta_H \setminus \Delta^*$ with T_i in the boundary. Thus by construction, (U_1, U_2) is a partition of the set of vertices $(V(\Omega_H) \setminus V(\Omega_{J'})) \cup \pi_{\rho_{J'}}((T_1 \cup T_2) \cap N(\rho_{J'}))$. Finally, for $i \in [2]$, fix the closed disk Δ_i in $\Delta_H \setminus \Delta^*$ such that $\Delta_i \cap (\mathsf{bd}(\Delta_H) \cup T_i) = \pi_{\rho^\star}^{-1}(U_i)$.

Defining the subgraphs contained in Δ_1 and Δ_2. Let L be the minimal subgraph of G such that $G = G' \cup L$. Note that L is a subgraph of H.

Claim 5.2.14. *$V(L) \cap V(G') \subseteq U_1 \cup U_2$.*

Proof of claim. Consider a vertex $x \in V(L) \cap V(G')$. By the minimality of L, the vertex x cannot be deleted from L and still have the property that $L \cup G' = G$. Thus, there exists an edge xy incident to x which is not contained in G' and consequently, xy is in H, the inner graph of C_7. Given that $x \in V(G')$, the vertex x is either in the outer graph of C_7 or in J'. If x is in the outer graph of C_7, but not in J', then $x \in V(\Omega_H) \setminus V(\Omega_{J'})$ and thus $x \in U_1 \cup U_2$. Thus, we may assume $x \in V(J')$. However, in this case as $(J', \Omega_{J'})$ is isolated in (H, Ω_H), it follows that $x \in \pi_{\rho_{J'}}((T_1 \cup T_2) \cap N(\rho_{J'}))$. We conclude that $x \in U_1 \cup U_2$ as claimed. $\diamond$

Claim 5.2.15. *There is no $U_1 - U_2$-path in L.*

Proof of claim. Assume otherwise that a $U_1 - U_2$-path R exists. R can be extended to a Ω-path R' using Q_1, Q_m, and C_7. Then R' along with $\{Q_2, ..., Q_{m-1}\}$ and C_7 forms a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{Q}$ up to level C_8, given that $m - 2 \geq k$. $\diamond$

Let K_1 be the union of all components of L which contain a vertex of U_1 and let K_2 be the union of all components of L which contain no vertex of U_1. Hence, by Claim 5.2.15, K_2 contains all components of L which contain a vertex of U_2.

Defining the new rendition ρ'. We are now ready to define the desired rendition $\rho' = (\Gamma', \mathcal{D}')$ of (G, Ω). Let $\rho^\star = (\Gamma^\star, \mathcal{D}^\star)$. Define $\mathcal{D}' = \{D \in \mathcal{D}^\star : \mathrm{int}(D) \subseteq \Delta \setminus (\Delta_1 \cup \Delta_2)\} \cup \{\Delta_1, \Delta_2\}$. The drawing Γ' is obtained from the restriction of $\Gamma^\star$ to $\Delta \setminus (\mathrm{int}(\Delta_1) \cup \mathrm{int}(\Delta_2))$ along with an arbitrary drawing of K_i in Δ_i such that the only points on the boundary of Δ_i are exactly the vertices of $U_i \cap V(K_i)$ for $i \in [2]$.

Defining the nests. Observe that for all $i \in [2, m-1]$, the path Q_i is contained in G' and the subgraph $Q_i \cup C_{7+i}$ contains exactly two cycles which are distinct from C_{7+i}. If we consider the track in $\rho^\star$ of each these two cycles, one bounds a disk in Δ which contains Δ_1 and the other bounds a disk which contains Δ_2. For $i \in [s'+1]$, we define C_i^1 (resp. C_i^2) to be the cycle contained in $Q_{i+1} \cup C_{7+i}$ (resp. $Q_{m-i} \cup C_{7+i}$) and distinct from C_{7+i} such that the disk bounded by the track of C_i^1 (resp. C_i^2) in $\rho^\star$ contains Δ_1 (resp. Δ_2). Observe that, for $i \in [2]$, $\mathcal{C}_i = (C_1^i, \ldots, C_{s'}^i)$ forms a nest around Δ_i. Additionally, note that $\mathcal{C}_1$ and $\mathcal{C}_2$ are disjoint given that they do not intersect $\{Q_i \mid i \in [s'+2, m-s'-1]\}$ and that $m - s' - 1 \geq s' + 4$. Moreover, $\mathcal{P}_1$ and $\mathcal{P}_2$ immediately validate the conditions of the statement.

Defining Δ_i'. Let Δ_i' be the closed subdisk bounded by the track of $C_{s'+1}^i$. Then $\rho_i = \rho'[\Delta_i']$ is indeed a cylindrical rendition with vortex $c_i = \Delta_i$. Note that Δ_1' and Δ_2' do not intersect given that $m \geq 2s' + 5$, and thus that the track of $\mathcal{Q}_{s'+3}$ is disjoint from both Δ_1' and Δ_2'. Additionally, for each $P \in \mathcal{P}_i$, $i \in [2]$, there is a unique subpath from $\Omega_{\mathcal{D}_i'}$ to $V(\sigma(c_i))$. Then the set $\mathcal{P}_i'$ of all such subpaths checks all constraint of the second item of the lemma. This concludes the proof. ☐

Finding a planar rendition of small depth and breadth

We now prove that a cylindrical rendition contains either a long-jump transaction, or a crosscap transaction, or has a rendition in the plane of small depth and small breadth.

We define recursively

$$f_{5.2.16}(k, r) = a_k f_{5.2.16}(k-1, r) + b_k$$

with $f_{5.2.16}(1, r) = 14$, $a_k = 6(r-1)(2k+1)$, and $b_k = 6(r-1)k(3k+5) + 6$. Hence,

$$f_{5.2.16}(k, r) = (\prod_{i=1}^{k} a_i) f_{5.2.16}(1, r) + \sum_{i=2}^{k} (\prod_{j+1}^{k} a_j) b_i = 2^{O(k \log(k \cdot r))}.$$

Theorem 5.2.16. *Let $k, r, s \in \mathbb{N}$ with $k \geq 1$ and $s, t \geq f_{5.2.16}(k, r)$. Let (G, Ω) be a society and $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω). Let $(\mathcal{C} = (C_1, C_2, \ldots, C_s), \mathcal{P})$ be a railed nest in ρ around c_0 of order (s, t). Then one of the following holds.*

(i) *There is an r-crosscap transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $f_{5.2.16}(k, r)$,*

(ii) *There is a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $f_{5.2.16}(k, r)$, or*

(iii) *(G, Ω) has a rendition in the disk of breadth at most $k-1$ and depth at most $f_{5.2.16}(k, r)$.*

Additionally, in case (iii), the closures of the vortex cells are pairwise disjoint.

Proof. Assume the theorem is false, and pick a counterexample to minimize k.

We fix the following values. Let $s' = f_{5.2.16}(k-1, r)$ and $t' = \lceil f_{5.2.16}(k-1, r)/2 \rceil$. Additionaly, let

$$\begin{aligned} m_3 &= 2s' + 5, \\ m_2' &= k(m_3 + 3k), \\ m_2 &= m_2' + 2t', \text{ and} \\ m_1 &= (m_2 - 1)(r - 1) + 1 \leq f_{5.2.16}(k, r)/6. \end{aligned}$$

Fix the cylindrical rendition ρ in a disk Δ. If (G, Ω) has no cross, then by Proposition 4.5.1 and given that $k \geq 1$, it satisfies (iii). Given a cross in (G, Ω), since $s \geq 11$ and $t \geq 14$, by Proposition 5.2.3, there is a cross in (G, Ω) that is coterminal with $\mathcal{P}$ up to level C_{11}. Given that $f_{5.2.16}(k, r) \geq 11$, (i) and (ii) hold in the case $r \leq 2$ and $k \leq 1$, respectively. We conclude that $r \geq 3$ and $k \geq 2$.

Step 1: Finding a crooked transaction. By Proposition 5.2.2, the society (G, Ω) has either a cylindrical rendition of depth $6m_1 \leq f_{5.2.16}(k, r)$, or a crooked transaction of cardinality m_1. In the former case, (iii) holds. Note we are using the fact that $m_1 \geq 4$. Hence, we can assume that there exists a crooked transaction $\mathcal{Q}_1$ of cardinality m_1. By Proposition 5.2.3, given that $s \geq 6m_1 \geq 2m_1 + 7$ and $t \geq 6m_1 \geq 4m_1 + 6$, we may assume that $\mathcal{Q}_1$ is coterminal with $\mathcal{P}$ up to level C_{2m_1+7}, and hence $C_{f_{5.2.16}(k,r)}$.

Step 2: Finding an unexposed planar transaction. Given that $m_1 \geq (m_2 - 1)(r - 1) + 1$, by Proposition 5.2.1, there is $\mathcal{Q}_2 \subseteq \mathcal{Q}_1$ that is either a crosscap transaction of cardinality r, or a planar transaction of cardinality m_2. In the first case, (i) holds, so we conclude that $\mathcal{Q}_2$ is a planar transaction.

Step 3: Creating rails. Let X_1 and X_2 be two disjoint segments of Ω such that $\mathcal{Q}_2$ is a transaction from X_1 to X_2. Let the elements of $\mathcal{Q}_2$ be enumerated $Q_1, \ldots, Q_{m_2}$ by the order in which their endpoints occur in X_1. Let $\mathcal{Q}'_2 = \{Q_i \mid i \in [t' + 1, t' + m'_2]\}$. Let $\mathcal{P}_1$ be the rail truncation of $\{Q_i \mid i \in [1, t']\}$ and $\mathcal{P}_2$ be the rail truncation of $\{Q_i \mid i \in [t' + m'_2 + 1, m'_2 + 2t']\}$. Hence, $\mathcal{P}_1$ and $\mathcal{P}_2$ both have $2t'$ elements.

Step 4: Finding a rural and isolated strip society. Apply Lemma 5.2.6 to the transaction $\mathcal{Q}'_2$ in (G, Ω) with the rendition ρ, $l' = m_1$, $l = m'_2$, and the nest $(C_{2m_1+7}, C_{2m_1+8}, \ldots, C_s)$. We can do so because $s - 2m_1 - 6 \geq 9$ and $m'_2 \geq k(m_3 + 3k)$. If we find a long-jump transaction of order k that is coterminal with $\mathcal{P}$ up to level C_{2m_1+15}, then we satisfy (ii) given that $f_{5.2.16}(k, r) \geq 2m_1 + 15$. Thus, we may assume that we find a transaction $\mathcal{Q}_3 \subseteq \mathcal{Q}'_2$ of size m_3 such that the $\mathcal{Q}_3$-planar strip society in (G, Ω) is rural and isolated.

Step 5: Splitting the vortex in two. We apply Lemma 5.2.13 to (G, Ω) with $m = m_3$, $l = 2t'$, railed nest $((C_{2m_1+7}, \ldots, C_s), \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3)$ where $\mathcal{P}_3$ is the rail truncation of $\mathcal{Q}_3$. We can do so because $t' \geq 1$, $m_3 \geq 2s' + 5 \geq k + 2$, and $s - 2m_1 - 6 \geq 4m_1 - 6 \geq 8ks' - 6 \geq s' + 8$. If there is a long-jump transaction of order k that is coterminal with $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$ up to level C_{2m_1+15}, then this transaction is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,r)}$ since $f_{5.2.16}(k, r) \geq 2m_1 + 15$. We conclude that there is a rendition ρ' of (G, Ω) of breadth two, with vortices c_1 and c_2, two disjoint ρ'-aligned disks Δ_1 and Δ_2 such that, for $i \in [2]$, $\rho'[\Delta_i]$ is a cylindrical rendition $\rho_i = (\Gamma_i, \mathcal{D}_i, c_i)$ of $(G_i, \Omega_i) = (\mathsf{inner}'_\rho(\Delta_i), \Omega_{\Delta_i})$ with a railed nest $(\mathcal{C}_i = (C^i_1, \ldots, C^i_{s'}), \mathcal{P}'_i)$ of order (s', t') such that, for each $P \in \mathcal{P}'_i$, P is a subpath of an element of $\mathcal{P}_i$ and, for each $j \in [s']$, $P \cap C^i_j = P \cap C_{j+\alpha}$, where $\alpha = 2m_1 + 21$. Additionally, $(C_{2m_1+14}, \ldots, C_s)$ is a nest around both c_1 and c_2, and there is a path $Q \in \mathcal{Q}_3$ such that Δ_1 and Δ_2 are contained in different connected components of $\Delta \setminus T$, where T is the track of Q in ρ'.

Step 6: Finding a contradiction. For $i \in [2]$, let k_i be the largest value such that there exists a k_i-long-jump transaction $\mathcal{R}_i$ in (G_i, Ω_i) that is coterminal with $\mathcal{P}'_i$ up to level $C^i_{s'}$.

Claim 5.2.17. *For $i \in [2]$, $k_i > 0$.*

Proof of claim. Let X'_1 and X'_2 be minimal segments of Ω such that $\mathcal{Q}_3$ is a linkage from X'_1 to X'_2. Let Y_1 and Y_2 be the two maximal segments of $\Omega \setminus (X'_1 \cup X'_2)$. Let $i \in [2]$. Since the planar transaction $\mathcal{Q}_3$ is a subset of the crooked transaction $\mathcal{Q}_1$, there exists a path $P_i \in \mathcal{Q}_1$ with an endpoint in Y_i that is crossed by another path $P'_i \in \mathcal{Q}_1$. If P_i crosses some $Q \in \mathcal{Q}_3$, then, given that the $\mathcal{Q}_3$-strip society in (G, Ω) is isolated, it follows that the other endpoint of P_i is in Y_2. But then, given that $m_3 \geq k$, we conclude that there is a long-jump transaction $\mathcal{Q} \subseteq \mathcal{Q}_1$ of order k in (G, Ω). Given that Q_1 is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,r)}$, (ii) holds, a contradiction. Hence, P_i crosses no $Q \in \mathcal{Q}_3$, and neither does P'_i. Hence, P_i and P'_i have both endpoints in Y_i. The pairs of paths (P_1, P'_1) and (P_2, P'_2) form a cross, one in (G_1, Ω_1) and the other in (G_2, Ω_2) since there is a path $Q \in \mathcal{Q}_3$ such that the interiors of Δ_1 and Δ_2 are contained in different connected components of $\Delta \setminus T$, where T is the track of Q in ρ'. Therefore, since $s' \geq 11$ and $t' \geq 14$, by Proposition 5.2.3, for $i \in [2]$, there is a cross, and hence a 1-long-jump transaction, in (G_i, Ω_i) that is coterminal with $\mathcal{P}'_i$ up to level C^i_{11}, and hence $C^i_{s'}$. We conclude that $k_i > 0$. ⋄

Claim 5.2.18. $k_1 + k_2 < k$.

Proof of claim. Given that $\mathcal{P}'_1$ and $\mathcal{P}'_2$ are disjoint linkages, whose paths are subpaths of $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively, that $(C_{2m_1+14}, \ldots, C_s)$ is a nest around both c_1 and c_2, and that for each $P \in \mathcal{P}'_i$ and $j \in [s']$, $P \cap C^i_j = P \cap C_{i+\alpha}$ by Lemma 5.2.13, we conclude that, for $i \in [2]$, $\mathcal{R}_i$ can be extended using $\mathcal{P}_i \subseteq \mathcal{Q}_1$ to a long-jump transaction $\mathcal{R}'_i$ of (G, Ω), and that $\mathcal{R}'_1$ and $\mathcal{R}'_2$ form either a (k_1, k_2)-double-jump transaction or, along with Q, an alternative (k_1, k_2)-double-jump transaction that is coterminal with $\mathcal{P}$ up to level $C_{s'+\alpha}$. Therefore, by Lemma 5.2.5, there is a $(k_1 + k_2)$-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $C_{s'+\alpha+1}$, and thus $C_{f_{5.2.16}(k,r)}$, since $f_{5.2.16}(k, r) \geq 6m_1 \geq s' + 2m_1 + 22$. Given that there is no k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,r)}$, we conclude that $k_1 + k_2 < k$. ⋄

By Claim 5.2.17 and Claim 5.2.18, we conclude that, for $i \in [2]$, $1 \leq k_i \leq k - 2$. By the choice of our counterexample to minimize k, it follows that one of (i)-(iii) must hold for (G_i, Ω_i) with parameters $k_i + 1$ and r. We can do so because $s', 2t' \geq f_{5.2.16}(k - 1, r, a)$. By the maximality of k_i, outcome (ii) does not hold.

If (i) holds, then there is a r-crosscap transaction in (G_i, Ω_i) that is coterminal with $\mathcal{P}'_i$ up to level $C^i_{f_{5.2.16}(k_i+1,r)}$. Then, given that for each $P \in \mathcal{P}'_i$ and $j \in [s']$, $P \cap C^i_j = P \cap C_{i+\alpha}$, we can extend $\mathcal{Q}_i$ to a crosscap transaction of (G, Ω) that is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k_i+1,r)+\alpha}$. Given that $f_{5.2.16}(k_i + 1, r) + \alpha \leq f_{5.2.16}(k_i + 2, r) \leq f_{5.2.16}(k, r)$, (i) thus holds for (G, Ω).

Hence, outcome (iii) holds for both (G_1, Ω_1) and (G_2, Ω_2). Thus, for $i \in [2]$, there is a rendition ρ'_i of (G_i, Ω_i) of breadth at most k_i and depth at most $f_{5.2.16}(k_i + 1, r)$. Given that $k_1 + k_2 < k$ by Claim 5.2.18, we conclude that (G, Ω) has a rendition of breadth at most $k_1 + k_2 \leq k - 1$ and depth at most $f_{5.2.16}(k - 1, r)$, by restricting ρ' to $\Delta \setminus (\mathsf{int}(\Delta_1) \cup \mathsf{int}(\Delta_2))$ and using ρ'_1 and ρ'_2 in the disks Δ_1 and Δ_2, respectively. In particular, given that Δ_1 and Δ_2 are disjoint, the closures of the vortex cells of this rendition are pairwise disjoint. This contradiction completes the proof. □

Finding a projective rendition of small depth and breadth

In the previous section, we proved that if a society has neither a long-jump nor a crosscap transaction, then it has a rendition in the plane of small breadth and depth. Our goal is now to only exclude a long-jump transaction to get a rendition in the projective plane of small breadth and depth.

We first borrow the following result from [195]. It essentially says that, if our cylindrical rendition with vortex c_0 contains a crosscap transaction $\mathcal{Q}$, then we can use $\mathcal{Q}$ to add a crosscap to the surface (that is, we are now in the projective plane), and find a new cylindrical rendition with vortex c_1 avoiding the crosscap.

Proposition 5.2.19 (Lemma 10.2, [195])**.** *Let $m, m', s, s', t \in \mathbb{N}$ with $s \geq s'+8$ and $m \geq m'+2s'+7$. Let (G, Ω) be a society and $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω) in the disk Δ. Let X_1, X_2 be disjoint segments of $V(\Omega)$. Let $\mathcal{C} = (C_1, \ldots, C_s)$ be a nest in ρ of size s and $\mathcal{P} = \{P_1, P_2, \ldots, P_t\}$ a linkage from X_1 to $\widetilde{c_0}$ orthogonal to $\mathcal{C}$. Let $\Sigma^\star$ be a surface homeomorphic to the projective plane minus an open disk obtained from Δ by adding a crosscap to the interior of c_0. If there exists an m-crosscap transaction $\mathcal{Q}$ in (G, Ω) orthogonal to $\mathcal{C}$ and disjoint from $\mathcal{P}$ such that every member of $\mathcal{Q}$ has both endpoints in X_2 and the $\mathcal{Q}$-strip society in (G, Ω) is isolated and rural, then there exists a subset $\mathcal{Q}'$ of $\mathcal{Q}$ of size m' and a rendition ρ' of (G, Ω) in $\Sigma^\star$ such that there exists a unique vortex $c' \in C(\rho')$ and the following hold:*

- (i) *$\mathcal{Q}'$ is disjoint from $\sigma(c')$,*
- (ii) *the vortex society of c' in ρ' has a cylindrical rendition $\rho_1 = (\Gamma_1, \mathcal{D}_1, c_1)$,*
- (iii) *every element of $\mathcal{P}$ has an endpoint in $V(\sigma_{\rho_1}(c_1))$,*
- (iv) *ρ_1 has a nest $\mathcal{C}' = (C'_1, \ldots, C'_{s'})$ of size s' such that $\mathcal{P}$ is orthogonal to $\mathcal{C}'$ and for every i, $1 \leq i \leq s'$, and for all $P \in \mathcal{P}$, $C'_i \cap P = C_{i+7} \cap P$,*
- (v) *for $i = 1, 2, \ldots, t$, let x_i be the endpoint of P_i in X_1, and let y_i be the last entry of P_i into c' (which exists), then if $x_1, x_2, \ldots, x_m$ appear in Ω in the order listed, then $y_1, y_2, \ldots, y_m$ appear on $\widetilde{c'}$ in the order listed,*
- (vi) *let Δ' be the open disk bounded by the track of C_s; then ρ restricted to $\Delta \setminus \Delta'$ is equal to ρ' restricted to $\Delta \setminus \Delta'$.*

We can finally prove our local structure theorem.

Theorem 5.2.20. *Let $k, r, s, t \in \mathbb{N}$ such that $s \geq f_{5.2.16}(k, m_1) + f_{5.2.16}(k, k+1) + 7$ and $t \geq f_{5.2.16}(k, m_1)$ where $m_1 = (6k+1) f_{5.2.16}(k, k+1) + 3k(4k+9)$. Let (G, Ω) be a society and $\rho = (\Gamma, \mathcal{D}, c_0)$ be a cylindrical rendition of (G, Ω) in the disk Δ. Let $(\mathcal{C} = (C_1, C_2, \ldots, C_s), \mathcal{P})$ be a railed nest in ρ around c_0 of order (s, t). Then one of the following holds.*

- (i) *There is a k-long-jump transaction in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $f_{5.2.16}(k, m_1)$, or*
- (ii) *(G, Ω) has a rendition in the plane of depth at most $f_{5.2.16}(k, m_1)$ or in the projective plane of depth at most $f_{5.2.16}(k, k+1)$, and of breadth at most $k-1$ in both cases*

Additionally, in case (ii), the closures of the vortex cells of the rendition are pairwise disjoint.

Proof. Assume that the theorem is false. We fix the following values. Let

$$\begin{aligned} s', t' &= f_{5.2.16}(k, k+1), \\ m_3 &= k + 2s' + 9, \\ m_2 &= 3k(m_3 + 3k), \text{ and} \\ m_1 &= m_2 + t'. \end{aligned}$$

Step 1: Finding a crosscap transaction. We apply Theorem 5.2.16 with $r = m_1$. We can do so because $s, t \geq f_{5.2.16}(k, m_1)$. Outcome (ii) and (iii) of Theorem 5.2.16 immediately imply outcome (i) and (ii) of the theorem respectively. Hence, there is a m_1-crosscap transaction $\mathcal{Q}_1$ in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $f_{5.2.16}(k, m_1)$.

Step 2: Separating $\mathcal{Q}_1$ into a crosscap transaction $\mathcal{Q}_2$ and rails $\mathcal{P}'$. Let X be a minimal segment of Ω containing both endpoints of every path in $\mathcal{Q}_1$. Let X_1 and X_2 be two disjoint segments contained in X such that X_2 contains both endpoints of m_2 paths of $\mathcal{Q}_1$; we call this set of paths $\mathcal{Q}_2$; and X_1 contains one endpoint of the t' paths in $\mathcal{Q}_1 \setminus \mathcal{Q}_2$. We can do so because $m_1 \geq m_2 + t'$. Let $\mathcal{P}'$ be the rail truncation of $\mathcal{Q}_1 \setminus \mathcal{Q}_2$. $\mathcal{P}'$ has size at least t'.

Step 3: Finding a rural and isolated strip society in $\mathcal{Q}_2$. Let X_1' and X_2' be the minimal segments contained in X_2 so that each path of $\mathcal{Q}_2$ has one endpoint in X_1' and the other in X_2'. We apply Lemma 5.2.6 with input the society (G, Ω), the nest $(C_{f_{5.2.16}(k,m_1)}, \dots, C_s)$, the crosscap transaction $\mathcal{Q}_2$, segments X_1' and X_2', $l = m_3$, and $l' = m_2$. We can do so because $s - f_{5.2.16}(k, m_1) + 1 \geq 9$ and $m_2 \geq 3k(m_3 + 3k)$. If there is a long-jump transaction of order k in (G, Ω) that is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,m_1)+9}$, then (i) holds given that $s \geq f_{5.2.16}(k, m_1) + 9$. Hence, there exists a transaction $\mathcal{Q}_3 \subseteq \mathcal{Q}_2$ of size m_3 such that the $\mathcal{Q}_3$-strip of (G, Ω) with respect to (X_1', X_2') is isolated and rural.

Step 4: Embedding the society in the projective plane. Let $\Sigma^\star$ be a surface homeomorphic to the projective plane minus an open disk obtained from Δ by adding a crosscap to the interior of c_0. We apply Proposition 5.2.19 to the society (G, Ω), segments X_1, X_2, nest $(C_{f_{5.2.16}(k,m_1)}, \dots, C_s)$, $\mathcal{P}'$, $\mathcal{Q}_3$, and $m' = k$. We can do so because $s - f_{5.2.16}(k, m_1) + 1 \geq s' + 8$ and $m_3 \geq k + 2s' + 7$. Thus, there exists a transaction $\mathcal{Q}_4 \subseteq \mathcal{Q}_3$ of size k, a rendition ρ' of (G, Ω) in $\Sigma^\star$, and a unique vortex $c_0' \in C(\rho')$ satisfying items (i)-(vi) of Proposition 5.2.19. In particular, $\mathcal{Q}_4$ is disjoint from $\sigma(c_0')$, and the vortex society (G_1, Ω_1) of c_0' in ρ' has a cylindrical rendition $\rho_1 = (\Gamma_1, \mathcal{D}_1, c_1)$ with nest $\mathcal{C}' = (C_1', ..., C_{s'}')$.

Step 5: Finding a contradiction. We apply Theorem 5.2.16 to (G_1, Ω_1) with the cylindrical rendition ρ_1, nest $\mathcal{C}' = (C_1', ..., C_{s'}')$, the truncation $\mathcal{P}''$ of $\mathcal{P}'$ in (G_1, Ω_1) for ρ_1, and $r = k + 1$. We can do so because $s', t' \geq f_{5.2.16}(k, k + 1)$. There are three cases.
Case 1: There is a long-jump transaction $\mathcal{Q}_5$ of order k in (G_1, Ω_1) that is coterminal with $\mathcal{P}''$ up to level $C'_{f_{5.2.16}(k,k+1)}$. By items (i)-(vi) of Proposition 5.2.19 and given that $\mathcal{P}''$ is the truncation of $\mathcal{P}'$, $\mathcal{Q}_5$ can extended to a long-jump transaction of order k in (G, Ω) that is coterminal with $\mathcal{P}'$ up to level $C_{f_{5.2.16}(k,k+1)+7}$, and hence with $\mathcal{P}$ up to level $C_{g(k)}$ given that $s \geq f_{5.2.16}(k, m_1) \geq f_{5.2.16}(k, k + 1) + 7$.
Case 2: there is a crosscap transaction $\mathcal{Q}_5$ in (G_1, Ω_1) of size $k+1$ that is coterminal with $\mathcal{P}''$ up to level $C'_{f_{5.2.16}(k,k+1)}$. Similarly to Case 1, $\mathcal{Q}_5$ can be extended to a crosscap transaction $\mathcal{Q}_6$ of size $k + 1$ in (G, Ω) that is coterminal with $\mathcal{P}'$ up to level $C_{f_{5.2.16}(k,k+1)+7}$. In particular, $\mathcal{Q}_4$ and $\mathcal{Q}_6$ are disjoint, are coterminal to $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,k+1)+7}$, $\mathcal{Q}_4$ has size k and $\mathcal{Q}_6$ has size $k + 1$. Therefore, there is a $(k, k + 1)$-klein transaction $\mathcal{Q}_4 \cup \mathcal{Q}_6$ that is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,k+1)+7}$. But then, by Lemma 5.2.5, there is a k-long-jump transaction that is coterminal with $\mathcal{P}$ up to level $C_{f_{5.2.16}(k,k+1)+7+k}$. (i) thus holds given that $f_{5.2.16}(k, m_1) \geq f_{5.2.16}(k, k+1)+7+k$.
Case 3: (G_1, Ω_1) has a rendition ρ_1' in the disk of breadth at most $k - 1$ and depth at most $f_{5.2.16}(k, k + 1)$, such that the closure of the vortex cells in ρ_1' are pairwise disjoint. Hence, (G, Ω) has a rendition in $\Sigma^\star$ of breadth at most $k-1$ and depth at most $f_{5.2.16}(k, k+1)$ by restricting ρ_1 to $\Delta \setminus (\mathsf{int}(c_0'))$ and using ρ_1' in c_0'. Trivially, the closure of the vortex cells in this rendition are pairwise disjoint. This contradiction completes the proof. □

5.2.2 From societies to a local structure theorem

In this section, we will prove our local structure theorem using the main result of previous section (Theorem 5.2.20). To apply Theorem 5.2.20, we need to find a cylindrical rendition in the input graph G. We do so using the Flat Wall theorem (which originates from [271]) which find a grid-like structure (a *wall*) with a vortex-free rendition, called a *flat wall*, in a graph G that excludes a big clique as a minor, after the removal of a small (apex) set $A \subseteq V(G)$. More specifically, we prove a new version of the flat wall theorem (see Section 5.2.2, Lemma 5.2.22), where we exclude a long-jump grid instead of a clique as a minor, which allows us to find a flat wall without needing to remove an apex set. It is then enough to make a vortex out of everything that is not part of the flat wall to obtain a cylindrical rendition. We can thus put everything together and prove that if G contains no big long-jump grid as a minor, then G has a Σ-decomposition δ of small breadth and depth, where Σ is the projective plane (see Section 5.2.2, Theorem 5.2.26). Additionally, we add another property on δ concerning *tangles*, that we define there.

Finding a flat wall

In the part, we adapt the Flat Wall theorem [194, 271] to long-jump grids (Theorem 5.2.25). The Flat Wall theorem essentially states that, given $k, r \in \mathbb{N}$ and a big enough wall W in a graph G, either G contains a K_k minor or there is a set A such that an r-subwall of W is a flat wall of $G - A$. We essentially prove here that given given $k, r \in \mathbb{N}$ and a big enough wall W in a graph G, either G contains a $\mathscr{J}_k$ minor or there is a r-subwall of W that is a flat wall of G.

The following is a statement of the Grid Theorem. While we will not explicitly use the Grid Theorem, we will, later on, make use of the existence of a function that forces the existence of a large wall in a graph with large enough treewidth.

Proposition 5.2.21 (Grid Theorem [65, 261])**.** *There exists a universal constant $c \geq 1$ such that for every $k \in \mathbb{N}$ and every graph G, if $\mathsf{tw}(G) \geq ck^{10}$, then G contains the $(k \times k)$-grid as a minor.*

Notice that any graph that contains a $(2k \times 2k)$-grid as a minor contains a k-wall as a subgraph. Let us prove our flat wall theorem when we exclude a long-jump grid as a minor.

Lemma 5.2.22. *Let $k, r \geq 1$ be integers, with r odd, let G be a graph, and let W be a $((r + 3k)k + 6k - 4, r + 8k - 4)$-wall in G. Then one of the following holds.*

- *Either G contains $\mathscr{J}_k$ minor grasped by W, or*
- *there is a flat wall W' in G, where W' is the tilt of an r-subwall of W.*

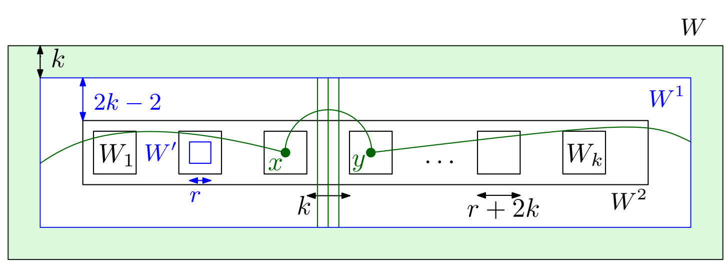

Figure 5.6: The walls $W_1, ..., W_k$ in W. In darkgreen is depicted the $\mathscr{J}_k$ minor of Claim 5.2.23.

Proof. Let W^1 be the $((r + 3k)k + 4k - 4, r + 6k - 4)$-wall obtained by removing the first k layers of W. Let W^2 be the $((r + 3k)k, r + 2k)$-wall obtained by removing the first $2k - 2$ layers of W^1. Let $a = (r + 3k)k$,and $b = r + 2k$. Let $P_1, ..., P_b$ (resp. $Q_1, ..., Q_a$) be the horizontal (resp. vertical) paths of W^2. Let $I'_1, ..., I'_k$ be intervals of length $r + 3k$ with union $[a]$. For $i \in [k]$, let I_i be obtained from I'_i by deleting the last k elements. Thus, $|I_i| = r + 2k$. For $i \in [k]$, we define W_i as the $(r+2k)$-subwall of W^2 obtained by removing all paths Q_j for $j \in [a] \setminus I_i$. Refer to Figure 5.6 for an illustration. Observe that the k first layers of W do not contain vertices from the W_i, $i \in [k]$, and, hence, form, along with horizontal and vertical subpaths, a railed nest $(\mathcal{C}, \mathcal{P})$ of order $(k, 2a+2b)$ (in light green in Figure 5.6).

Let B be a W-bridge in G with at least one attachment in $V(W_i)$. Let B' be obtained from B by deleting all its attachments that do not belong to $V(W_i)$. Let H_i be the graph obtained from the union of W_i and all graphs B' defined as above.

Claim 5.2.23. *If there are indices i and j with $i < j$ such that H_i and H_j share a vertex, then G contains a $\mathscr{J}_k$ minor grasped by W.*

Proof of claim. Assume that there are indices i and j with $i < j$ such that H_i and H_j share a vertex. Then there exists a W-bridge with an attachment $x \in V(W_i)$ and $y \in V(W_j)$. Therefore there exists a W-path P in G with endpoints x and y. Then, the union of P, the k vertical paths Q_l for $l \in I'_i \setminus I_i$, the railed nest $(\mathcal{C}, \mathcal{P})$, and an appropriate horizontal subpath from x (resp. y) to $\mathcal{C}$ in W, contains $\mathscr{J}_k$ (grasped by W) as a minor. ⋄

In this case, we directly conclude, so we may assume that the subgraphs H_i are pairwise disjoint. Let C_i be the perimeter of W_i for $i \in [k]$. Let $N_i = V(C_i) \cap N_G(V(G) \setminus V(H_i))$. Let Ω_i be the cyclic ordering of the vertices of N_i as they appear in C_i. If, for each $i \in [k]$, (H_i, Ω_i) contains a cross, then the nested-crosses grid $\mathscr{N}\mathscr{C}_k^{k+1}$ is a minor of G. By Lemma 5.2.4, $\mathscr{J}_k$ is hence a minor of G (grasped by W). In this case, we can directly conclude, so we may assume that there exists an $i \in [k]$ such that (H_i, Ω_i) do not have a cross. Then, by Proposition 4.5.1, the society (H_i, Ω_i) is rural. It implies that W_i is a flat wall of H_i.

We now want to obtain a flat wall of G. Let W'' be the r-wall obtained from W_i by deleting the k first layers of W_i. By Proposition 4.6.6, there is a flat wall W' of H_i such that W' is a tilt of W''. Let (X', Y) be a separation of H_i and (P, C) be a choice of pegs and corners certifying that W' is flat wall of H_i, with $V(W) \subseteq Y$. Let $X = X' \cup (V(G) \setminus V(H_i))$. Let us show that (X, Y) along with (P, C) certifies that W' is a flat wall of G. For this, it is enough to prove the following.

Claim 5.2.24. *(X, Y) is a separation of G or G contains $\mathscr{J}_k$ grasped by W has a minor.*

Proof of claim. Suppose that (X, Y) is not a separation of G, and thus that there are $x \in X \setminus Y$ and $y \in Y \setminus X$ that are adjacent. Given that (X', Y) is a separation of H_i, it implies that $x \notin V(H_i)$ and $y \in V(H_i)$.

Assume that $y \in V(W_i) \setminus V(W')$. Let D be the perimeter of W'. Then there is a path from y to $V(C_i)$ disjoint from $V(D)$, contradicting the fact that $V(C_i) \subseteq X'$, $y \in Y$, and $X' \cap Y \subseteq V(D)$. Hence, $y \notin V(W_i) \setminus V(W')$.

The edge joining x and y belongs to a W-bridge B of G, and hence, x is an attachment of B outside W_i. Thus, there is a W-path with one endpoint x and the other $y' \in V(W')$. If $x \in V(W^1)$, then, by a similar argument to Claim 5.2.23, G contains $\mathscr{J}_k$ (grasped by W) as a minor. Otherwise, $x \in V(W) \setminus V(W^1)$. But then, there are k paths from $W_i \setminus V(W')$ plus $2k-2$ paths from $W^1 \setminus W^2$ separating x from y'. Hence, G contains the alternative jump grid $\hat{\mathscr{J}}_{2k}^{3k-2}$, and thus $\mathscr{J}_k$ (grasped by W) by Lemma 5.2.4 as a minor. ⋄

Thus, the separation (X, Y) witnesses that W' is a flat wall in G. □

Given that a (a, b)-wall is a subset of an $\mathcal{O}(\sqrt{ab})$-wall, Lemma 5.2.22 immediately implies the following.

Theorem 5.2.25. *Let $k, r \geq 1$ be integers, with r odd, let G be a graph, and let W be a $f_{5.2.25}(k, r)$-wall in G, where $f_{5.2.25}(k, r) = \mathcal{O}(\sqrt{k}(r + k))$. Then one of the following holds.*

- *Either G contains $\mathscr{J}_k$ minor grasped by W, or*
- *there is a flat wall W' in G, where W' is the tilt of an r-subwall of W.*

Tangles

To create the tree decomposition of the global structure theorem, we require tangles, first introduced by Robertson and Seymour in [268].

Tangles. Let G be a graph and k be a positive integer. We denote by $\mathcal{S}_k$ the collection of all separations (A, B) of order less than k in G. An *orientation* of $\mathcal{S}_k$ is a set $\mathcal{O}$ such that for all $(A, B) \in \mathcal{S}_k$ exactly one of (A, B) and (B, A) belongs to $\mathcal{O}$. A *tangle* of order k in G is an orientation $\mathcal{T}$ of $\mathcal{S}_k$ such that for all $(A_1, B_1), (A_2, B_2), (A_3, B_3) \in \mathcal{T}$, we have $A_1 \cup A_2 \cup A_3 \neq V(G)$.

Tangles induced by walls. Let W be a k-wall in a graph G and (A, B) be a separation of order strictly less than k. Then exactly one of $A \setminus B$ or $B \setminus A$ contains a row and a column of W. Let $\mathcal{T}_W$ be the orientation of $\mathcal{S}_k$ where $(A, B) \in \mathcal{T}_W$ if and only if $B \setminus A$ contains a row and a column of W. Then it is easy to observe that $\mathcal{T}_W$ is a tangle, which we call the *tangle induced by W*.

Truncation. Let $\mathcal{T}' \subsetneq \mathcal{T}$ be a tangle which is properly contained in the tangle $\mathcal{T}$. We say that $\mathcal{T}'$ is a *truncation* of $\mathcal{T}$. This implies in particular that the order of $\mathcal{T}'$ is smaller that the order of $\mathcal{T}$.

Flat walls in a Σ-decomposition. Let W be a wall in a graph G. We say that W in *flat* is a Σ-decomposition δ of G if there exists a δ-aligned disk Δ such that

- $\pi(N(\delta) \cap \mathsf{bd}(\Delta)) \subseteq V(D(W))$,
- if S is the collection of corners and 3-branch vertices of W that are not in $\mathsf{ground}(\delta)$, then, for any $c \in C(\delta)$, there exists at most one $v \in S$ such that $v \in V(\sigma(c))$,
- no cell $c \in C(\delta)$ with $c \subseteq \Delta$ is a vortex, and
- $W - V(D(W))$ is a subgraph of $\bigcup_{c \subseteq \Delta} \sigma(c)$.

In the next result, we reformulate the local structure theorem in the form we need to prove the global structure theorem in the next section.

Theorem 5.2.26. *There exist functions $f_{5.2.26} \colon \mathbb{N}^2 \to \mathbb{N}$, $d_{5.2.26} \colon \mathbb{N} \to \mathbb{N}$ such that, for every choice of non-negative integers k, r with odd $r \geq 3$ and every graph G with an $f_{5.2.26}(k, r)$-wall W, one of the following holds*

(i) *G contains the long-jump grid of order k as a minor, or*

(ii) *G has a Σ-decomposition δ of breadth at most $k - 1$ and depth at most $d_{5.2.26}(k)$ such that the closures of the vortex cells in δ are pairwise disjoint. Moreover, Σ is either the sphere or the projective plane and there exists a wall of height at least r which is flat in δ and whose tangle is a truncation of the tangle induced by W.*

Moreover, $f_{5.2.26}(k, r) = \max\{2^{\mathcal{O}(k \log k)}, \mathcal{O}(\sqrt{k}(r + k))\}$ and $d_{5.2.26}(k) = 2^{\mathcal{O}(k \log k)}$.

Proof. We set $m_1 = (6k+1)f_{5.2.16}(k,k+1) + 3k(4k+9)$, $\ell = f_{5.2.16}(k,m_1) + f_{5.2.16}(k,k+1) + 7$, and $d_{5.2.26}(k) = f_{5.2.16}(k,m_1)$. We also set r_2 to be the smallest odd integer bigger than $\max\{r, f_{5.2.16}(k,m_1)/4\}$, $r_1 = 2(\ell+1) + r_2$, and $f_{5.2.26}(k,r) = f_{5.2.25}(k,r_1)$. By Theorem 5.2.25, given that $f_{5.2.26}(k,r) = f_{5.2.25}(k,r_1)$, either G contains $\mathscr{J}_k$ as a minor, in which case we conclude, or there is a flat wall W_1 in G, where W_1 is the tilt of an r_1-subwall of W. In the latter case, let W_2 be the central r_2-subwall of W_1. Given that W_1 is a flat wall in G, there is a separation (X,Y) of G and a cyclic ordering Ω of the vertices of $X \cap Y$ witnessing the flatness of W_1. In particular, $(G[Y],\Omega)$ has a vortex-free rendition $\rho = (\Gamma,\mathcal{D})$ in a disk Δ. Let T be the track of the perimeter of W_2. $\mathbb{S}_0 \setminus T$ is the union of two disks whose closure is Δ_1 and Δ_2 respectively. We assume without loss of generality that $W_2 - V(D(W_2))$ is a subgraph of $\bigcup_{c \subseteq \Delta_2} \sigma(c)$. Notice that $\pi(N(\rho) \cap \mathsf{bd}(\Delta_2)) \subseteq V(D(W_2))$, given that T is the track of $D(W_2)$ and the boundary of Δ_2. Let G' be the graph obtained from G after removing, for each $c \in C(\rho)$ with $c \subseteq \Delta_2$, the edges of $\sigma_\rho(c)$ and the vertices of $\sigma_\rho(c) - \pi_\rho(T \cap N(\rho))$. Let Ω' be the cyclic ordering of the vertices of $\pi_\rho(T \cap N(\rho))$ with the cyclic order induced by the perimeter of W_2. We construct a cylindrical rendition $\rho' = (\Gamma',\mathcal{D}',c_0)$ of (G',Ω') in the disk Δ_1 as follows. We set $\mathcal{D}' = \{c_0\} \cup \{c \in \mathcal{D} \mid c \subseteq \Delta_1\}$, where $c_0 = \mathbb{S}_0 \setminus \Delta$ with $\sigma(c_0) = G[X]$ and $\pi_{\rho'}(\tilde{c}_0) = X \cap Y$. We define Γ' to be obtained from the restriction of Γ to Δ_1 by drawing $G[X] - Y$ arbitrarily in c_0 and adding the appropriate edges with $\pi_{\rho'}(\tilde{c}_0)$. Given that, for $i \in [2,\ell+1]$, the i-th layer of W_1 is drawn in $\Delta_1 \setminus c_0$, ρ' is a cylindrical rendition of (G',Ω') with a railed nest $(\mathcal{C},\mathcal{P})$ in ρ around c_0 of order $(\ell, 4r_2)$. See Figure 5.7 for an illustration.

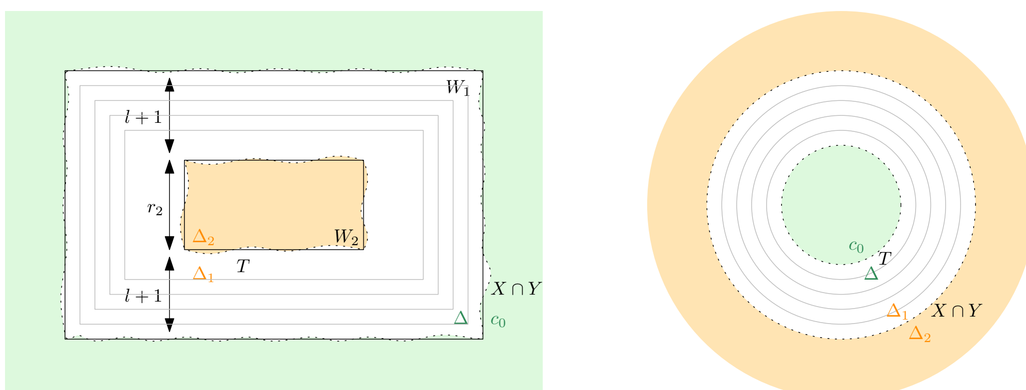

Figure 5.7: Illustration for the proof of Theorem 5.2.26. ρ is the vortex-free rendition in the disk Δ witnessing that W_1 is a flat wall. ρ' is the cylindrical rendition composed of all but the orange disk Δ_2, where the vortex c_0 is composed of all but the disk Δ.

Given that $\ell = f_{5.2.16}(k,m_1) + f_{5.2.16}(k,k+1) + 7$ and that $4r_2 \geq f_{5.2.16}(k,m_1)$, by Theorem 5.2.20, either (a) (G',Ω') contains a k-long-jump transaction that is coterminal with $\mathcal{P}$ up to level $f_{5.2.16}(k,m_1)$, or (b) (G',Ω') has a rendition in the plane of depth at most $f_{5.2.16}(k,m_1)$ or in the projective plane of depth at most $f_{5.2.16}(k,k+1)$, and of breadth at most $k-1$ in both cases, such that the closures of the vortex cells are pairwise disjoint. Given that $\ell - f_{5.2.16}(k,m_1) + 1 \geq k$, it implies in case (a) that $\mathscr{J}_k$ is a minor of G, so we can conclude. In case (b), by combining the restriction of ρ to Δ_2 and ρ', we get a Σ-decomposition δ of breadth at most $k-1$, where either Σ is the sphere and δ has depth at most $f_{5.2.16}(k,m_1) = d_{5.2.26}(k)$, or Σ is the projective plane and δ has depth at most $f_{5.2.16}(k,k+1) \leq d_{5.2.26}(k)$, such that the closure of the vortex cells are pairwise disjoint.

Given that W_2 is a subwall of W, its tangle is obviously a truncation of the tangle induced by W. Moreover, from the flatness of W_1, we easily derive that W_2 is flat in δ. Hence the result. $\square$

If we want to exclude both a long-jump grid and a crosscap grid, then we get a similar result.

Theorem 5.2.27. *There exist functions $f_{5.2.27}\colon \mathbb{N}^3 \to \mathbb{N}$, $d_{5.2.27}\colon \mathbb{N}^2 \to \mathbb{N}$ such that, for every choice of non-negative integers k, c, r with odd $r \geq 3$ and every graph G with an $f_{5.2.27}(k,r)$-wall W, one of the following holds*

(i) *G contains the long jump grid of order k as a minor, or*

(ii) *G contains the crosscap grid of order c as a minor, or*

(iii) *G has a $\mathbb{S}_0$-decomposition δ of breadth at most $k-1$ and depth at most $d_{5.2.27}(k,c)$ such that the closures of the vortex cells in δ are pairwise disjoint. Moreover, there exists a wall of height at least r which is flat in δ and whose tangle is a truncation of the tangle induced by W.*

Moreover, $f_{5.2.27}(k,r) = \max\{2^{\mathcal{O}(k\log(k\cdot c))}, \mathcal{O}(\sqrt{k}(r+k))\}$ and $d_{5.2.27}(k) = 2^{\mathcal{O}(k\log(k\cdot c))}$.

Proof. The proof is exactly the same as for Theorem 5.2.26, but using Theorem 5.2.16 instead of Theorem 5.2.20, and setting $\ell = d_{5.2.27}(k) = f_{5.2.16}(k,c)$, r_2 to be the smallest integer bigger than $\max\{r, f_{5.2.16}(k,c)/4\}$, $r_1 = 2(\ell+1) + r_2$, and $f_{5.2.27}(k,c,r) = f_{5.2.25}(k,r_1)$. □

5.2.3 The global structure theorem

In this section, we will prove our global structure theorem (Theorem 2.1.2, and more precisely Theorem 5.2.39) using the local structure theorem (Theorem 5.2.26) proved in the previous section. For now, we know that if G contains no big long-jump grid as a minor, then G has a Σ-decomposition δ of small breadth and depth, where Σ is the projective plane.

Instead of the depth, the parameter we actually care about on vortices is their *width*. A graph with a Σ-decomposition of breadth and width at most k (with some additional properties) is said to be *k-almost embeddable* in Σ. We prove in Theorem 5.2.36 that, if G excludes a long-jump grid as minor, then it has a tree decomposition $\mathcal{T} = (T, \beta)$ such that the torso at each node has an almost embedding on the projective plane of small breadth and width. Imagine that all of $V(G)$, and thus the Σ-decomposition δ of small breadth and depth, is originally in the root r of T. Then, essentially, most of what is inside each vortex c_v is pushed out to a child t_v of r. Then, in the root, we obtain that each vortex has now small width instead of small depth, and it remains to recurse on the children of r. This is the very standard "local to global" approach, used for instance in [87, 88, 101, 195, 303]. However, there is a catch here. To our knowledge, the structure theorem proved in this chapter is the first structure theorem with vortices but *no apices* (note that the structure theorem of [268] for singly-crossing graphs has no apices, but also no vortices). Indeed, usually, one proves that, the torso of $\mathcal{T}$ at each node is almost embeddable in some surface after the removal of a bounded set of vertices (the *apices*), with sometimes some additional properties. These apices are very useful, in particular to hide among them the set X of vertices in the intersection of r and t_v. By definition of the torso, we need to make a clique out of X, which we can do here safely without destroying the almost embeddability of the rest of the bag. In our case however, we have no apices. Hence, we develop in Theorem 5.2.32 a new technique to go from the local to the global structure theorem *in the absence of apices*. This technique will be explained more in detail later and, as we already mentioned, it works because the surface we consider is either the plane or the projective plane, but that it would not work on other surfaces given that we do not know how to handle cycles that do not bound a disk.

Finally, we easily derive from Theorem 5.2.36 our global structure theorem (Theorem 2.1.2, and more precisely Theorem 5.2.39) in terms of identifications: the set of vertices in vortex cells has bounded bidimensionality, so we can identify each vortex separately to obtain an embedding in the projective plane.

From local to global

The proof of the global structure theorems, providing the upper bounds for our main results, follows a well-established strategy, that was formalized in [88]. This strategy allows us to prove a slightly stronger "rooted" version of the global structure theorem (Theorem 5.2.32). The main advantage of this stronger version is that it allows for a straightforward proof by induction.

Let us first give some definition before sketching how this well-established strategy usually works.

Linear decompositions. A *linear decomposition* of a society (G, Ω) is a labeling $v_1, \ldots, v_\ell$ of $V(\Omega)$, such that $v_1, \ldots, v_n$ occur in that order on Ω, and subsets $(X_1, \ldots, X_\ell)$ such that

- for each $i \in [\ell]$, $v_i \in X_i \subseteq V(G)$,
- $\bigcup_{i\in[\ell]} X_i = V(G)$ and, for each $uv \in E(G)$, there exists $i \in [\ell]$ such that $\{u, v\} \in X_i$, and
- for each $x \in V(G)$, the set $\{i \mid x \in X_i\}$ is an interval in $[\ell]$.

The *width* of a linear decomposition is $\max_{i\in[\ell]} |X_i|$.

It is not hard to see that every society with a linear decomposition of adhesion at most d has depth at most $2d$. For the converse, we have the following result.

Proposition 5.2.28 ([195, 265])**.** *Let $d \in \mathbb{N}$. Every society of depth at most d has a linear decomposition of adhesion at most d.*

Almost embeddings. Let G be a graph and Σ be a surface. An *almost embedding* of G in Σ of *breadth* b and *width* d is a Σ-decomposition δ of G such that there is a set $C_0 \subseteq C(\delta)$ of size at most b containing all vortex cells of δ such that:

- no vertex of G is drawn in the interior of a cell of $C(\delta) \setminus C_0$ and,
- for each vortex cell, there exists a linear decomposition of its vortex society of width at most d (and for each non-vortex cell $c \in C_0$, $|V(\sigma(c))| \leq d$).

C_0 is called the *vortex set* of δ.

Well-linked sets. Let $\alpha \in [2/3, 1)$. Moreover, let G be a graph and $X \subseteq V(G)$ be a vertex set. A set $S \subseteq V(G)$ is said to be an α*-balanced separator* for X if for every component C of $G - S$ it holds that $|V(C) \cap X| \leq \alpha|X|$. Let k be a non-negative integer. We say that X is a (k, α)*-well-linked set* of G if there is no α-balanced separator of size at most k for X in G.

Given a (k, α)-well-linked set X of G we define

$$\mathcal{T}_X \coloneqq \{(A, B) \in \mathcal{S}_{k+1}(G) \mid |X \cap B| > \alpha|X|\}.$$

It is not hard to see that $\mathcal{T}_S$ is a tangle of order $k + 1$ in G.

We need an algorithmic way to find, given a well-linked set, a large wall whose tangle is a truncation of the tangle induced by the well-linked set. This is done in [302] by algorithmatising a proof Kawarabayashi, Wollan, and Thomas from [195].

Proposition 5.2.29 (Thilikos and Wiederrecht [302] (see Theorem 3.4.))**.** *Let $k \geq 3$ be an integer and $\alpha \in [2/3, 1)$. There exist universal constants $c_1, c_2 \in \mathbb{N} \setminus \{0\}$, and an algorithm that, given a graph G and a (c_1k^{20}, α)-well-linked set $X \subseteq V(G)$ computes in time $2^{\mathcal{O}(k^{c_2})}|V(G)|^2|E(G)|\log(|V(G)|)$ a k-wall $W \subseteq G$ such that $\mathcal{T}_W$ is a truncation of $\mathcal{T}_X$.*

The rooted version of a structure theorem is usually stated along the lines of: *Let G, H be graphs and $X \subseteq V(G)$ be a set of small size. Then either G contains H as a minor, or there is a rooted tree decomposition (T, β, r) of G such that the torso at each node has an almost embedding in Σ after removing a small apex set A, and such that $X \subseteq \beta(r)$.*

Obviously, if $X = \emptyset$, then this is the global structure theorem. X essentially corresponds to vertices inherited from a parent bag in the induction, from which we will make a clique to obtain the torso. The proof of such a result goes as follows. If there is a balanced separator S in G for X, then we inductively find a rooted tree decomposition $\mathcal{T}_C = (T_C, \beta_C, t_C)$ for each connected component C of $G - S$ with $X_C = X \cap C \cup S$. Then $\mathcal{T} = (T, \beta, r)$ is the tree decomposition where the children of r are the nodes t_C, $\beta(r) = S \cup X$, and the restriction of $\mathcal{T}$ to the subtree rooted at t_C is $\mathcal{T}_C$. Otherwise, X is a well-linked set from which we can derive a wall (Proposition 5.2.29) whose tangle agrees with the tangle of X. Then, we can apply the local structure theorem (for us Theorem 5.2.26) to find an apex set A (for us $A = \emptyset$) such that $G - A$ has a Σ-decomposition δ of small breadth and depth at most d. For each non-vortex cell c, we recurse on $G_c^1 = \sigma(c)$ with X_c^1 that is the union of $X \cap \sigma(c)$ and the boundary A_c^1 of the cell, to find a tree decomposition $\mathcal{T}_c^1$. For each vortex cell c, we fix a linear decomposition $(Y_1, \dots, Y_\ell)$ of adhesion at most its depth. Then, for each $i \in [\ell]$, we recurse on $G_c^i = G[Y_i]$ with X_c^i being the union of $X \cap Y_i$ and the set A_c^i composed of its adhesion with its neighbors as well as the ith vertex of the cyclic ordering Ω_c, and we obtain a tree decomposition $\mathcal{T}_c^i$. Then, we put all those tree decompositions together, that we attach to the root r with $\beta(r)$ that is the union of A, X, and the sets A_c^i, to obtain a tree decomposition $\mathcal{T} = (T, \beta, r)$ of G. It remains to prove that the torso of $\mathcal{T}$ at each node t has an almost embedding in Σ after removing a small apex set A of small depth and small width. This is immediate for $t \in V(T)$ that is not a child of r. For r, torsifying corresponds to making a clique out of each X_c^1. For each c, i, let us add $X_c^i \setminus A_c^i$ to the apex set. The size of the apex set increases by at most $|X|$. Now, it is enough to prove that making a clique out of each A_c^i does not destroy the almost embedding. For non-vortex cells, we have $|A_c^1| \leq 3$, so making a clique trivially does not destroy the almost embedding. For vortex cells, A_c^i is now a bag of the linear decomposition, so the width does not increase after the torsification. Additionally, A_c^i has size at most $2d + 1$, so the width of a vortex is at most $2d + 1$. Hence, we have an almost embedding of $\beta(r)$ of small breadth and width at most $2d + 1$ after removing the apex set $A + X$. Finally, for each child t_c^i of r, after removing an apex set B_c^i, we already have an almost embedding of the torso of $\mathcal{T}_c^i$ at t_c^i, where $\mathcal{T}_c^i$ is a tree decomposition of G_c^i. To make it an almost embedding of the torso of $\mathcal{T}$ at t_c^i, we need to add all edges between the vertices of X_c^i (which is the adhesion of $\beta(r)$ and $\beta(t_c^i)$), which might destroy the almost embedding. To handle this problem, it is enough to remove X_c^i from the almost embedding, that is to add X_c^i to the apex set. Hence, we have an almost embedding of the torso of $\mathcal{T}$ at t_c^i of small breadth and width at most $2d + 1$ after removing the apex set $B_c^i \cup X_c^i$, which concludes the proof.

In our case, we cannot add X_c^i to the apex set. That is, we need to argue that, even if we add edges between the vertices of X_c^i, *it does not destroy* the almost embedding too much, in such a way that, by creating new vortices, there is still an almost embedding of bounded breadth and bounded depth. Before going further, let us define a stellation in a graph and the torso of a set in a graph (to distinguish with that torso of a tree decomposition at a node).

Stellation and torso. Let G be a graph and $\mathcal{S}$ be a collection of subsets of $V(G)$. We denote by $G_{\mathcal{S}}^{\star}$ the graph obtained from G by adding, for each $S \in \mathcal{S}$, a vertex v_S adjacent to the vertices in S. The vertices v_S are called *stellation vertices*. Let $X \subseteq V(G)$. Recall that the *torso* of X in G, denoted by $\mathsf{torso}(G, X)$ is the graph derived from the induced subgraph $G[X]$ by turning $N_G(V(C))$ into a clique for each connected component C of $G - X$. We denote by $G_{\mathcal{S}}^{\circ}$ the graph obtained from G by turning each $S \in \mathcal{S}$ into a clique. In other words, $G^{\circ}_{\mathcal{S}}$ is the torso of $V(G)$ in $G^{\star}_{\mathcal{S}}$. See Figure 5.8 for an illustration.

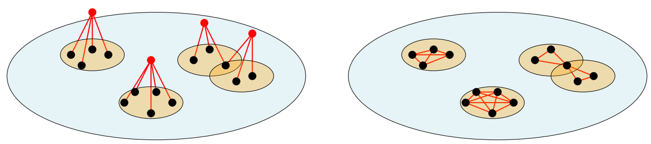

Figure 5.8: Illustration of $G^{\star}_{\mathcal{S}}$ and $G^{\circ}_{\mathcal{S}}$, where the sets in $\mathcal{S}$ are depicted in orange, and the new vertices and edges are depicted in red.

Note that we have two definitions of torso. One is torso of a set $X \subseteq V(G)$ in a graph G defined just above. The other one (see Section 4.3) is, given a tree decomposition $\mathcal{T} = (T, \beta)$ of G, the torso of $\mathcal{T}$ at node t. This second notion is stronger, in the sense that the torso of $\mathcal{T}$ at node t is the graph obtained from $G[\beta(t)]$ by making a clique out of each set $\beta(t) \cap \beta(t')$ with t' adjacent to t, while the torso of $\beta(t)$ in G makes a clique out of subsets of the sets $\beta(t) \cap \beta(t')$. However, we can prove the following.

Lemma 5.2.30. *If a graph G has treewidth k, then there is always a tree decomposition $\mathcal{T} = (T, \beta)$ of G of width k such that the torso of $\mathcal{T}$ at node t is exactly the torso of $\beta(t)$ in G.*

Moreover generally, let $\mathcal{H}$ be a hereditary graph class and suppose that G admits a tree decomposition $\mathcal{T}' = (T', \beta')$ of width k such that the torso $\mathcal{T}'$ at each node is in $\mathcal{H}$. Then, there is a tree decomposition $\mathcal{T} = (T, \beta)$ of G of width k such that the torso G_t of $\mathcal{T}$ at node t is exactly the torso of $\beta(t)$ in G and that $G_t \in \mathcal{H}$.

Proof. While there is $t \in V(T')$ such that the torso of $\mathcal{T}'$ at node t is different from the torso of $\beta'(t)$ in G, we modify the tree decomposition $\mathcal{T}'$ as follows. If the two notion of torso differ for t, then this means that there is a neighbor t' of t such that several connected components $C_1, \ldots, C_\ell$ of $G - \beta'(t)$, with $\ell \geq 2$, contain vertices of $\beta'(t')$, meaning that we add more edges in the torso of $\mathcal{T}$ at t (where we make a clique out of all of $\beta'(t) \cap \beta'(t')$) than in the torso of $\beta'(t)$ in G (where we only make a clique out of each $N_G(V(C_i)) \subseteq \beta'(t) \cap \beta'(t')$). We may modify the tree decomposition by removing the subtree $T'_{t'}$ rooted at t' (where we assume that the root is t), and adding instead ℓ copies $T^1_{t'}, \ldots, T^\ell_{t'}$ with $\beta'(u_i) = \beta'(u) \cap V(C_i)$ for $u \in V(T_{t'})$ and u_i its copy in $T^i_{t'}$, and joining t to each copy of t'. In this new tree decomposition, for both notions of the torso, we make a clique out of the entirety of $\beta'(t) \cap \beta'(t_i)$ for $i \in \ell$. For each $u \in V(T'_t)$, if the torso G_u at u is in $\mathcal{H}$, then the torso at u_i is equal to $G_u \cap V(C_i)$, which is also in $\mathcal{H}$ given that $\mathcal{H}$ is hereditary. Hence, by proceeding as such by induction, we obtain the desired tree decomposition. □

Theorem 5.2.32 is the rooted version of our global theorem and it essentially goes as follows:

> Let G, H be graphs (H is a long-jump grid), $X \subseteq V(G)$ be a set of small size, and $\mathcal{S}$ be a collection of subsets of X. Then either $G^{\star}_{\mathcal{S}}$ contains H as a minor, or there is a rooted tree decomposition $\mathcal{T} = (T, \beta, r)$ of $G^{\circ}_{\mathcal{S}}$ such that, for each node t of T, the torso of $\beta(t)$ in $G^{\circ}_{\mathcal{S}}$ has an almost embedding in Σ and such that $X \subseteq \beta(r)$.

If $X = \emptyset$, the above gives our global structure theorem by Lemma 5.2.30. Let us give intuition on X and $\mathcal{S}$. As previously, X corresponds to vertices that are present in the parent bag inherited from the induction. Given that we now work with the torso of $\beta(t)$ and not the torso of $\mathcal{T}$ at t, we will not make a clique out of all of X. Instead, we make a clique out of a subset of X if it neighbors some connected component with respect to the parent. This is what $\mathcal{S}$ represents.

Let us sketch the proof of our result. As previously, if X has a balanced separator, we easily conclude, so we may assume that there is a wall in G, and thus in $G^\star_{\mathcal{S}}$. We now apply the local structure theorem on $G^\star_{\mathcal{S}}$ and conclude that, if $G^\star_{\mathcal{S}}$ is H-minor-free, then there is a Σ-decomposition of $G^\star_{\mathcal{S}}$ of small breadth and depth at most d. For each cell c, we define G^i_c, A^i_c, and X^i_c as previously. We additionally define $\mathcal{S}^i_c$ to be the collection of all sets $N_{G^\star_{\mathcal{S}}}(C)$ such that C is a connected component of $G^\star_{\mathcal{S}} - G^i_c$. Hence, we can recurse on G^i_c, X^i_c, and S^i_c. $(G^i_c)^\star_{\mathcal{S}^i_c}$ is a minor of $G^i_{\mathcal{S}}$ (obtained by contracting each connected component C of $G^\star_{\mathcal{S}} - G^i_c$ to a point, so if H is a minor of $(G^i_c)^\star_{\mathcal{S}^i_c}$, it is also a minor of $G^\star_{\mathcal{S}}$. So we can assume that we found a tree decomposition $\mathcal{T}^i_c$ respecting the criteria. Then we define the tree decomposition $\mathcal{T} = (T, \beta, r)$ just as before. It remains to prove that there is an almost embedding of the torso of $\beta(t)$ in $G^\circ_{\mathcal{S}}$ of small breadth and depth for each $t \in V(T)$. This is immediate for $t \neq r$ by induction. The difficult part is to prove so for r. We currently have an almost embedding in $G^\star_{\mathcal{S}}$ of small breadth and width at most $2d + 1$ (each cell containing vertices of X is added to the vortex set). To obtain the torso of $\beta(r)$ in $G^\circ_{\mathcal{S}}$, we need to make a clique out of each X^i_c, as well as each $S \in \mathcal{S}$. Making a clique out of each X^i_c does not destroy the embedding. Moreover, for each S, if the corresponding stellation vertex v_S is in the interior of a cell c, then its neighborhood S is in $\sigma(c)$, so making a clique out of S does not destroy the embedding. The problem is when v_S is on the boundary of at least two cells. Then, it has neighbors in different cells between which we need to add an edge, hence destroying the almost embedding. The idea is to create a new vortex containing all cells with v_S on its boundary (whose number can be assumed to be bounded by $|X|$). The problem is that another stellation vertex $v_{S'}$ could possibly be on the boundary of this new vertex. So, we need to add all cells with $v_{S'}$ on its boundary to the new vortex, and so on. Two sets $S, S' \in \mathcal{S}$ whose stellation vertices are on the boundary of the same cell are said to be adjacent, which allows us to talk about connected components of $\mathcal{S}$. What we will show (using topological arguments) is that we can create a vortex c_Y for each connected component Y of $\mathcal{S}$ such that, if $S \in Y$, then v_S is in the interior of c_Y, and thus S is in $\sigma(c_Y)$, allowing us to make a clique out of S safely. These new vortices can be chosen to be distinct and to have width bounded by a function of $|X|$ and d. Hence, we find the desired almost embedding: instead of increasing the size of the apex set, what we grow is the width and the breadth of the embedding.

Before proving the rooted version of the global structure theorem, let us prove the following results, that essentially says that the number of stellation vertices (here $|S|$) is bounded by $|X|$.

Given a graph G embedded in a surface Σ such that the faces of G are disks, the *degree* of a face of G is the number of edges bounding the face (counted with multiplicity).

Lemma 5.2.31. *Let G be a graph embeddable in the projective plane, and (X, S) be a partition of $V(G)$ such that S is an independent set and $|X| \geq 1$. Then $|\{N_G(s) \cap X \mid s \in S\}| \leq 6|X| - 4$.*

Proof. Without loss of generality, we can assume that the vertices in S have pairwise distinct neighborhood. Then it is enough to prove that $|S| \leq 6|X| - 5$. Note that, if $|X| = 1$, then $|S| \leq 1 = 6|X| - 5$, and, if $|X| = 2$, then $|S| \leq 3 \leq 6|X| - 5$. Hence, we can assume that $|X| \geq 3$. If G is not planar (resp. planar), then there is an embedding of G in the projective plane (resp. the sphere), such that each face of G is a disk.

Let d_i be the number of vertices in S of degree i, for $i \in [0, 2]$, and $d_{\geq 3}$ be the number of vertices of S of degree at least three. Given that S is an independent set and that the vertices of S have distinct neighborhoods, we have $d_0 \leq 1$ and $d_1 \leq |X|$. Let G' be the simple graph obtained from G by removing each vertex of S and

- for each vertex of S of degree two, adding an edge between its neighbors, and

- for each vertex of S of degree three or more with neighbors $x_1, \ldots, x_\ell$ appearing in this order in the embedding, adding an edge between x_1 and x_{i+1} for $i \in [\ell]$ (modulo ℓ).

G' is a graph with vertex set X that is embeddable in the projective plane. In particular, d_2 is bounded by the number of edges of G', and $d_{\geq 3}$ is bounded by the number of faces of G'.

Let v be the number of vertices, e be the number of edges, and f be the number of faces of G'. Let us compute the sum s of the degree of the faces of G'. Given that each face has degree at most three when $|V(G')| \geq 3$, we have $s \geq 3f$. Additionally, given that each edge bounds two faces (or maybe once, but then it counts twice), we have $s = 2e$. Therefore, $3f \leq 2e$. By Euler's formula for the projective plane (resp. the sphere) [87], we also have $v + f - e = 1$ (resp. $v + f - e = 2$). Therefore, we deduce that $e \leq 3v - 3$ and $f \leq 2v - 2$. Therefore, $|S| = d_0 + d_1 + d_2 + d_{\geq 3} \leq 1 + |X| + (3|X| - 3) + (2|X| - 2) \leq 6|X| - 4$. ☐

We finally prove the rooted version of our global structure theorem.

Theorem 5.2.32. *There exists functions $f_{5.2.32}, b_{5.2.32}, w_{5.2.32} \colon \mathbb{N} \to \mathbb{N}$ such that for every positive integer k, every graph G, every set $X \subseteq V(G)$ of size at most $3f_{5.2.32}(k) + 1$, and every collection $\mathcal{S}$ of subsets of X, either*

1. *$G^{\star}_{\mathcal{S}}$ contains the long jump grid of order k as a minor, or*
2. *there exists a rooted tree decomposition (T, r, β) of $G^{\circ}_{\mathcal{S}}$ where*
   (a) *$X \subseteq \beta(r)$,*
   (b) *(T, β) has adhesion at most $3f_{5.2.32}(k) + 1$, and*
   (c) *for every $t \in V(T)$, $\mathsf{torso}(G^{\circ}_{\mathcal{S}}, \beta(t))$ has an almost embedding in the projective plane of breadth at most $b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$.*

Moreover, $f_{5.2.32}(k), w_{5.2.32}(k), b_{5.2.32}(k) = 2^{\mathcal{O}(k \log k)}$.

Proof. Let c_1 be the constant from Proposition 5.2.29. We set

$$\begin{aligned} f_{5.2.32}(k) &\coloneqq c_1 \cdot \left(f_{5.2.26}(k, 3)\right)^{20}, \\ w_{5.2.32}(k) &\coloneqq (24 f_{5.2.32}(k) + 4) \cdot (2 d_{5.2.26}(k) + 1), \text{ and} \\ b_{5.2.32}(k) &\coloneqq 3 f_{5.2.32}(k) + k. \end{aligned}$$

We prove the claim by induction on $|V(G) \setminus X|$.

If G is small. In case $|V(G)| \leq 3f_{5.2.32}(k) + 1$, we may select T to be the tree on one vertex, say t, and set $\beta(t) \coloneqq V(G)$. The resulting tree decomposition (T, β) of G, which is also a tree decomposition of $G^{\circ}_{\mathcal{S}}$, trivially meets the requirements of our assertion given that $3f_{5.2.32}(k) + 1 \leq w_{5.2.32}(k)$. Hence, we may assume that $|V(G)| > 3f_{5.2.32}(k) + 1$.

If X is small. Moreover, if $|X| \leq 3f_{5.2.32}(k)$, we may now select an arbitrary vertex $v \in V(G) \setminus X$ and set $X' \coloneqq X \cup \{v\}$. It follows that $|X'| \leq 3f_{5.2.32}(k) + 1$ and $|V(G) \setminus X| > |V(G) \setminus X'|$. Hence, by applying the induction hypothesis to G, X', and $\mathcal{S}$, we obtain either the long jump grid of order k as a minor of $G^{\star}_{\mathcal{S}}$, and are therefore done, or we obtain a rooted tree decomposition (T, r, β) of $G^{\circ}_{\mathcal{S}}$ meeting requirements *(a)*, *(b)*, and *(c)*. In particular, we have $X \subsetneq X' \subseteq \beta(r)$ and are therefore done. Thus, we may also assume that $|X| = 3f_{5.2.32}(k) + 1$.

If there is $X' \subseteq X$ such that $G - X'$ is disconnected. Let $X' \subseteq X$ and let $H_1, \ldots, H_\ell$ be the components of $G - X'$. If $\ell \geq 2$, then we may define a rooted tree decomposition (T, r, β) as follows. For each $i \in [\ell]$, let G_i be the graph induced by $X' \cup V(H_i)$. Let $\mathcal{S}_i$ be the collection of all the sets $N_{G^\star_\mathcal{S}}(V(C))$ such that C is a connected component of $G^\star_\mathcal{S} - V(G_i)$. Note that $(G_i)^\star_{\mathcal{S}_i}$ is a minor of $G^\star_\mathcal{S}$, given that it is obtained from $G^\star_\mathcal{S}$ by contracting each component of $G^\star_\mathcal{S} - V(G_i)$ to a single vertex. Moreover, for each $S \in \mathcal{S}_i$, $S \subseteq X \cap V(G_i)$. Given that $|V(G_i) \setminus X| < |V(G) \setminus X|$, by applying the induction hypothesis to G_i, $X \cap V(G_i)$, and $\mathcal{S}_i$, we obtain either that $\mathscr{J}_k$ is a minor of $(G_i)^\star_{\mathcal{S}_i}$, and thus of $G^\star_\mathcal{S}$, and therefore we are done, or we obtain a rooted tree decomposition (T_i, r_i, β_i) of $(G_i)^\circ_{\mathcal{S}_i}$ meeting requirements *(a)*, *(b)*, and *(c)*. Let T be the tree with root r obtained from the disjoint union of the trees T_i and the vertex r by joining r_i to r for every $i \in [\ell]$. We set $\beta(r) \coloneqq X$, and, for each $i \in [\ell]$ and $t \in V(T_i)$, we set $\beta(t) \coloneqq \beta_i(t)$. Requirements *(a)* and *(b)* are trivially satisfied. Moreover, given that $3f_{5.2.32}(k) + 1 \leq w_{5.2.32}(k)$, $\mathsf{torso}(G^\circ_\mathcal{S}, \beta(r))$ has an almost embedding in the plane composed of a single vortex of width at most $w_{5.2.32}(k)$, hence meeting requirement *(c)*. Finally, notice that, for $t \in V(T_i)$, the torso of $\beta(t)$ in $G^\circ_\mathcal{S}$ is equal to the torso of $\beta_i(t)$ in $(G_i)^\circ_{\mathcal{S}_i}$, hence meeting requirement *(c)*. Therefore (T, r, β) is indeed a rooted tree decomposition of $G^\circ_\mathcal{S}$ as required by the assertion. So we may assume that $G - X'$ is connected for any $X' \subseteq X$. In particular, G is connected.

If there exists a balanced separator for X. Suppose there exists a 2/3-balanced separator S of size at most $f_{5.2.32}(k)$ for X.

In this case let $H_1, \ldots, H_\ell$ be the components of $G - S$ and, for each $i \in [\ell]$, let $X'_i \coloneqq (X \cap V(H_i)) \cup S$. It follows that

$$\begin{aligned}|X'_i| &\leq \frac{2}{3}(3f_{5.2.32}(k) + 1) + |S| \\ &\leq 2f_{5.2.32}(k) + f_{5.2.32}(k) \\ &\leq 3f_{5.2.32}(k).\end{aligned}$$

Now, for each $i \in [\ell]$, if $|X'_i \cup V(H_i)| \leq d_{5.2.32}(k)$, we set $X_i \coloneqq X'_i \cup V(H_i)$ and we say that H_i is a *leaf*.

Otherwise, we select an arbitrary vertex $v_i \in V(H_i) \setminus X'_i$ and set $X_i \coloneqq X'_i \cup \{v_i\}$. Observe that $|X_i| \leq 3f_{5.2.32}(k) + 1$ in this case. For every $i \in [\ell]$ for which H_i is *not* a leaf, let $G_i = G[X_i \cup V(H_i)]$. Let also $\mathcal{S}_i$ be the collection of all the sets $N_{G^\star_\mathcal{S}}(V(C))$ such that C is a connected component of $G^\star_\mathcal{S} - V(G_i)$. This implies that $(G_i)^\star_{\mathcal{S}_i}$ can be obtained from $G^\star_\mathcal{S}$ by contracting each component of $G^\star_\mathcal{S} - V(G_i)$ to a single vertex, and thus, $(G_i)^\star_{\mathcal{S}_i}$ is a minor of $G^\star_\mathcal{S}$. Notice that the elements in $\mathcal{S}_i$ are subsets of X_i. We have that $|V(H_i) \setminus X_i| < |V(G) \setminus X|$ and thus, by applying the induction hypothesis to G_i, X_i, and $\mathcal{S}_i$, we obtain either that $\mathscr{J}_k$ is a minor of $(G_i)^\star_{\mathcal{S}_i}$, and thus of $G^\star_\mathcal{S}$, and therefore we are done, or we obtain a rooted tree decomposition (T_i, r_i, β_i) of $(G_i)^\circ_{\mathcal{S}_i}$ meeting requirements *(a)*, *(b)*, and *(c)*. For every $i \in [\ell]$ where H_i is a leaf we define such a rooted tree decomposition (T_i, r_i, β_i) by setting T_i to be the tree with a single vertex r_i and $\beta_i(r_i) \coloneqq X_i$.

Now let us define a rooted tree decomposition (T, r, β) for G as follows. Let T be the tree with root r obtained by taking the disjoint union of the vertex r and the trees T_i, and joining r_i to r for all $i \in [\ell]$. For every $i \in [\ell]$ and $t \in V(T_i)$, we set $\beta(t) \coloneqq \beta_i(t)$, and we set $\beta(r) \coloneqq X \cup S$. Note that $|\beta(r)| \leq 4f_{5.2.32}(k) + 1 \leq w_{5.2.32}(k)$, and that, for each $i \in [\ell]$ such that H_i is *not* a leaf, for each $t \in V(T_i)$, $\mathsf{torso}(G^\circ_\mathcal{S}, \beta(t)) = \mathsf{torso}((G_i)^\circ_{\mathcal{S}_i}, \beta_i(t))$. Then it is straightforward to check that (T, r, β) is indeed a rooted tree decomposition of $G^\circ_\mathcal{S}$ as required by the assertion. Hence, we may assume that there is no 2/3-balanced separator of size at most $f_{5.2.32}(k)$ for X.

Local structure theorem. If there is no 2/3-balanced separator of size at most $f_{5.2.32}(k)$ for X, then X is $(f_{5.2.32}(k), 2/3)$-well-linked in G, and thus also in $G^\star_{\mathcal{S}}$. By Proposition 5.2.29, given that $f_{5.2.32}(k) \coloneqq c_1 \cdot \left(f_{5.2.26}(k,3)\right)^{20}$, it implies that $G^\star_{\mathcal{S}}$ contains a $f_{5.2.26}(k,3)$-wall W such that $\mathcal{T}_W$ is a truncation of $\mathcal{T}_X$. Then, by Theorem 5.2.26, this implies that, either $\mathscr{J}_k$ is a minor of $G^\star_{\mathcal{S}}$, in which case we conclude, or $G^\star_{\mathcal{S}}$ has a Σ-decomposition $\delta = (\Gamma, \mathcal{D})$ of breadth at most $k-1$ and depth at most $d_{5.2.26}(k)$, where Σ is the projective plane, and there exists a wall W' of height at least three which is flat in δ and whose tangle is a truncation of the tangle induced by W. Additionally, the closure of the vortex cells of δ are pairwise disjoint. Let $\mathcal{C}_v$ be the set of vortex cells of δ.

Without loss of generality, we may assume that, for each cell $c \in C(\delta) \setminus \mathcal{C}_v$, and each distinct $u, v \in \pi_\delta(\tilde{c})$, there is a path from u to v whose internal vertices are in $\sigma_\delta(c) - \pi_\delta(\tilde{c})$. If not, then, if $|\tilde{c}| = 2$, then $\sigma_\delta(c)$ is disconnected and c can be divided into two cells c_u and c_v such that $\pi_\delta(\tilde{c}_u) = \{u\}$ and $\pi_\delta(\tilde{c}_v) = \{v\}$. And if $\pi_\delta(\tilde{c}) = \{u, v, w\}$, then w is a cut vertex of $\sigma_\delta(c)$ and c can be divided into two cells c_u and c_v such that $\pi_\delta(\tilde{c}_u) = \{u, w\}$ and $\pi_\delta(\tilde{c}_v) = \{v, w\}$.

Without loss of generality, we may also assume that, for each ground vertex $v \in \mathsf{ground}(\delta)$, there is $c \in C(\delta) \setminus \mathcal{C}_v$ such that $v \in \pi_\delta(\tilde{c})$ and $N_{\sigma_\delta(c)}(v) \neq \emptyset$. Indeed, suppose that is not the case. Then, given that G is connected, for any $c \in C(\delta)$ such that $v \in \pi_\delta(\tilde{c})$, c is a vortex cell. Given that the closure of the vortex cells are pairwise disjoint, v is thus drawn on the boundary of a unique cell $c \in \mathcal{C}_v$. Then, we can draw v in the interior of c instead of its boundary: It does not increase the depth of the vortex.

For each $\mathcal{R} \subseteq \mathcal{S}$, let $V_{\mathcal{R}}$ be the set of all stellation vertices v_S such that $S \in \mathcal{R}$. Let $\mathcal{S}^{\mathsf{g}}$ be the set of $S \in \mathcal{S}$ such that $v_S \in \mathsf{ground}(\delta)$. Let $X_1 \coloneqq X \cap \mathsf{ground}(\delta)$, X_2 be the set of vertices in X drawn in the interior of non-vortex cells, and X_3 be the set of vertices in X drawn in the interior of vortex cells. Notice that (X_1, X_2, X_3) is a partition of X. We have the following bound of the number of stellation vertices on the ground.

Claim 5.2.33. *If $\mathcal{S}^{\mathsf{g}} \neq \emptyset$, then $|\mathcal{S}^{\mathsf{g}}| \leq 6|X_1 \cup X_2| - 5$.*

Proof of claim. Let $\mathcal{C}_X$ be the set of cells $c \in C(\delta) \setminus \mathcal{C}_v$ such that $X_2 \cap \sigma_\delta(c) \neq \emptyset$. Let F be the graph with vertex set the union of X_1, $V_{\mathcal{S}^{\mathsf{g}}}$ and a vertex v_c for each $c \in C_X$, and edge set the edges of $G^\star_{\mathcal{S}}$ with both endpoints in $X_1 \cup V_{\mathcal{S}^{\mathsf{g}}}$ and, for each $c \in C_X$ and each $v_S \in \pi_\delta(\tilde{c}) \cap V_{\mathcal{S}^{\mathsf{g}}}$, an edge between v_c and v_S. By construction, F is embeddable in the projective plane. Let $X' \coloneqq X_1 \cup \{v_c \mid c \in C_X\}$. $V_{\mathcal{S}^{\mathsf{g}}}$ is an independent set of $G^\star_{\mathcal{S}}$ and thus of F. Additionally, by the assumptions on δ, for each $v_S \in V_{\mathcal{S}^{\mathsf{g}}}$, there is $c \in C(\delta) \setminus \mathcal{C}_v$ such that $v_S \in \pi_\delta(\tilde{c})$ with $N_{\sigma_\delta(c)}(v_S) \neq \emptyset$. Therefore, there is $x \in (X_1 \cup X_2) \cap S$ such that $x \in V(\sigma_\delta(c))$. Thus, $N_F(v_S) \cap X' \neq \emptyset$. We thus conclude, by Lemma 5.2.31 applied to $V_{\mathcal{S}^{\mathsf{g}}}$ and X', that $|\mathcal{S}^{\mathsf{g}}| = |V_{\mathcal{S}^{\mathsf{g}}}| \leq 6|X'| - 5 \leq 6|X_1 \cup X_2| - 5$. $\diamond$

Construction of the tree decomposition. For every non-vortex cell $c \in C(\delta) \setminus \mathcal{C}_v$, we set $G_c^1 = \sigma_\delta(c)$, $A_c^1 = \pi_\delta(\tilde{c})$, and $X_c^1 = (X \cap V(\sigma_\delta(c))) \cup A_c^1$. $(A, B) = (V(\sigma_\delta(c)), V(G^\star_{\mathcal{S}}) \setminus (V(\sigma_\delta(c)) \setminus A_c^1))$ is a separation of $G^\star_{\mathcal{S}}$ of order $|A_c^1| \leq 3 \leq f_{5.2.32}(k)$. Given that W' is flat in δ, and in particular that at most one 3-branch vertex of W' is in $\sigma_\delta(c) - \pi_\delta(\tilde{c})$, it follows that $A \setminus B$ does not contain a row and a column of W', and thus $(A, B) \in \mathcal{T}_{W'} \subseteq \mathcal{T}_X$. Therefore, $|X_c^1| \leq |X \cap V(\sigma_\delta(c))| + 3 \leq \frac{1}{3}|X| + 3 \leq 3f_{5.2.32}(k)$.

For every vortex cell $c \in \mathcal{C}_v$, let $(Y_1, \ldots, Y_\ell)$ be a linear decomposition of the vortex society $(\sigma_\delta(c), \Omega_c)$ of c of adhesion at most $d_{5.2.26}(k)$ (it exists by Proposition 5.2.28), with the vertices of $V(\Omega_c)$ labeled $v_1, \ldots, v_\ell$. For $i \in [\ell]$, let $A_c^i = (Y_i \cap Y_{i-1}) \cup (Y_i \cap Y_{i+1}) \cup \{v_i\}$ where $Y_0 = Y_{\ell+1} = \emptyset$. For each $i \in [\ell]$, we set G_c^i to be the graph induced by Y_i and $X_c^i = (X \cap Y_i) \cup A_c^i$. $(A, B) = (Y_i, V(G^\star_{\mathcal{S}}) \setminus (Y_i \setminus A_c^i))$ is a separation of $G^\star_{\mathcal{S}}$ of order $|A_c^i| \leq 2d_{5.2.26}(k) + 1 \leq f_{5.2.32}(k)$. Given that W' is flat in δ, and in particular that there is no vortex in the disk where the interior of W' is drawn, it follows that

$A \setminus B$ does not contain a row and a column of W', and thus $(A, B) \in \mathcal{T}_{W'} \subseteq \mathcal{T}_X$. Therefore, $|X \cap V(\sigma_\delta(c))| \leq \frac{1}{3}|X|$, and thus that $|X_c^i| \leq \frac{1}{3}|X| + 2d_{5.2.26}(k) + 1 \leq 3f_{5.2.32}(k)$.

For every cell $c \in C(\delta)$ and each i, we set $H_c^i := G_c^i - V_{\mathcal{S}}$ and $Z_c^i := X_c^i - V_{\mathcal{S}}$. We also set $\mathcal{S}_c^i$ to be the collection of all the sets $N_{G_{\mathcal{S}}^\star}(V(C))$ such that C is a connected component of $G_{\mathcal{S}}^\star - V(H_c^i)$. This implies that $(H_c^i)_{\mathcal{S}_c^i}^\star$ can be obtained from $G_{\mathcal{S}}^\star$ by contracting each component of $G_{\mathcal{S}}^\star - V(H_c^i)$ to a single point, and thus, $(H_c^i)_{\mathcal{S}_c^i}^\star$ is a minor of $G_{\mathcal{S}}^\star$. Notice that, for each $R \in \mathcal{S}_c^i$, $R \subseteq Z_c^i$.

We define a rooted tree decomposition (T, r, β) of G as follows. We define $\beta(r)$ to be the union of the sets Z_c^i, for all cells $c \in C(\delta)$ and all i. For every cell $c \in C(\delta)$ and each i such that $V(H_c^i) \setminus Z_c^i \neq \emptyset$, there exits $v_c^i \in V(H_c^i) \setminus Z_c^i$. Let $Z_c'^i = Z_c^i \cup \{v_c^i\}$. $|Z_c'^i| \leq 3f_{5.2.32}(k) + 1$ and $|V(H_c^i) \setminus Z_c'^i| < |V(G) \setminus X|$, so we can apply the induction hypothesis to H_c^i, $Z_c'^i$, and $\mathcal{S}_c^i$. We obtain either that $\mathscr{J}_k$ is a minor of $(H_c^i)_{\mathcal{S}_c^i}^\star$, and thus of $G_{\mathcal{S}}^\star$, and therefore we are done, or we obtain a rooted tree decomposition $(T_c^i, r_c^i, \beta_c^i)$ of $(H_c^i)_{\mathcal{S}_c^i}^\star$ meeting requirements *(a)*, *(b)*, and *(c)*. T is obtain from the union of r and the trees T_c^i by joining r to each r_c^i. For each $t \in V(T_c^i)$, we set $\beta(t) = \beta_c^i(t)$.

It is straightforward to check that (T, r, β) meet requirements *(a)* and *(b)*, and that each $t \in V(T) \setminus \{r\}$ meet requirement *(c)* (given that, for $t \in T_c^i$, $\mathsf{torso}(G_{\mathcal{S}}^\circ, \beta(t)) = \mathsf{torso}((G_c^i)_{\mathcal{S}_c^i}^\circ, \beta_c^i(t))$). It remains to prove that $\mathsf{torso}(G_{\mathcal{S}}^\circ, \beta(r))$ has an almost embedding in the projective plane of breadth at most $b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$. The difficulty is that we need to make a clique out of S for each $S \in \mathcal{S}$. If all vertices of S are drawn in the same cell, that is, if v_S is drawn in the interior of a cell, then it does not change much. However, if v_S is a ground vertex, then making a clique out of S destroys the Σ-decomposition. Therefore, we need to find a Σ-decomposition $\delta^\star$ such that each stellation vertex is drawn in the interior of a cell.

If there is no stellation vertex on the ground. If $\mathcal{S}^{\mathsf{g}} = \emptyset$, then $\mathsf{ground}(\delta) \subseteq \beta(r)$ and, for each $S \in \mathcal{S}$, there is a cell $c \in C(\delta)$ such that $S \subseteq V(\sigma_\delta(c))$. Then, we can construct a Σ-decomposition $\delta^\star$ of $\mathsf{torso}(G_{\mathcal{S}}^\circ, \beta(r))$ from the Σ-decomposition δ of $G_{\mathcal{S}}^\star$ by keeping only the vertices of $\beta(r)$ and making a clique out of each Z_c^i for $c \in C(\delta) = C(\delta^\star)$ and out of each S for each $S \in \mathcal{S}$. Let C_0 be the union of $\mathcal{C}_v$ and the cells of $C(\delta^\star) \setminus \mathcal{C}_v$ that have a vertex drawn on their interior. Notice that, by the definition of the sets X_c^1, the only vertices that can be drawn in the interior of non-vortex cells are vertices of X_2. Therefore, $|C_0| \leq k - 1 + |X_2| \leq b_{5.2.32}(k)$. For each $c \in C_0 \setminus \mathcal{C}_v$, $|V(\sigma_{\delta^\star}(c))| \leq 3 + |X_2|$. Let $c \in \mathcal{C}_v$. Remember that the vortex society $(\sigma_\delta(c), \Omega_c)$ of c in δ has a linear decomposition $(Y_1, \ldots, Y_\ell)$ of adhesion at most $d_{5.2.26}(k)$. For $i \in [\ell]$, let $Y_i' := (Y_i \cup X) \cap V(\sigma_{\delta^\star}(c)) = A_c^i \cup (X_3 \cap V(\sigma_{\delta^\star}(c)))$. Then $(Y_1', \ldots, Y_\ell')$ is a linear decomposition of the vortex society $(\sigma_{\delta^\star}(c), \Omega_c)$ in $\delta^\star$ of width at most $2d_{5.2.26}(k) + 1 + |X_3| \leq w_{5.2.32}(k)$. Therefore, $\delta^\star$ is a Σ-embedding with vortex set C_0 of breadth at most $b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$. Hence, we now assume that $\mathcal{S}^{\mathsf{g}} \neq \emptyset$.

If ground vertices are all stellation vertices. If $V_{\mathcal{S}^{\mathsf{g}}} = \mathsf{ground}(\delta)$, then we can define $\delta^\star$ to be the Σ-decomposition of $\mathsf{torso}(G_{\mathcal{S}}^\circ, \beta(r))$ composed of a unique vortex cell c with $V(\sigma_{\delta^\star}(c)) = \beta(r)$ and one arbitrary vertex on the boundary. Let us compute the width of the vortex society of c. Given that all vertices in $\mathsf{ground}(\delta)$ are stellation vertices, (X_2, X_3) is a partition of X. Note that $\sum_{c \in \mathcal{C}_v} |V(\Omega_c)| \leq |\mathsf{ground}(\delta)| = |\mathcal{S}^{\mathsf{g}}|$. Remember that the boundary of vortex cells of δ are pairwise disjoint. This implies that there is an injection between the bags in the linear decomposition $(Y_1, \ldots, Y_\ell)$ of vortex cells in δ and the vertices in $\mathcal{S}^{\mathsf{g}}$. Also, for each bag Y_i,

$|Y_i \cap \beta(r)| = |(Y_i \cap Y_{i-1}) \cup (Y_i \cap Y_{i+1})| \leq 2d_{5.2.26}(k)$. Therefore,

$$\begin{aligned}|\beta(r)| &\leq |X_2| + |X_3| + \sum_{c \in \mathcal{C}_v} |V(\Omega_c)| \cdot 2d_{5.2.26}(k)\\ &\leq |X| + |\mathcal{S}^{\mathsf{g}}| \cdot 2d_{5.2.26}(k)\\ &\leq 3f_{5.2.32}(k) + 1 + (6(3f_{5.2.32}(k)+1) - 5) \cdot 2d_{5.2.26}(k)\\ &\leq (36f_{5.2.32}(k) + 2) \cdot d_{5.2.26}(k) + 3f_{5.2.32}(k) + 1\\ &\leq w_{5.2.32}(k).\end{aligned}$$

Therefore, $\delta^\star$ is a Σ-embedding with vortex set C_0 of breadth at most $1 \leq b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$. We now assume that $\mathsf{ground}(\delta) \setminus V_{\mathcal{S}^{\mathsf{g}}} \neq \emptyset$.

Connected component of stellation vertices and its boundary. For $S, S' \in \mathcal{S}^{\mathsf{g}}$, we say that S and S' are *adjacent* if v_S and $v_{S'}$ are drawn on the boundary of the same cell of δ. We say that S and S' are *connected* if there is a sequence $S_0 = S, S_1, \ldots, S_{\ell-1}, S_\ell = S'$ such that, for each $i \in [\ell]$, S_{i-1} and S_i are adjacent. Hence, a *connected component* Y of $\mathcal{S}^{\mathsf{g}}$ is a maximal size set of elements of $\mathcal{S}^{\mathsf{g}}$ that are pairwise connected. Let us show that, for each connected component Y of $\mathcal{S}^{\mathsf{g}}$, we can replace the cells containing a stellation vertex v_S for $S \in Y$ in their boundary by a vortex, such that each stellation vertex v_S, for $S \in Y$, is drawn in the interior of the vortex.

Let Y be a connected component of $\mathcal{S}^{\mathsf{g}}$. It exists given that $\mathcal{S}^{\mathsf{g}} \neq \emptyset$. We call *interior* of Y the set of points $x \in N(\delta)$ such that $\pi_\delta(x) \in V_{\mathcal{S}^{\mathsf{g}}}$. Points in the interior of Y are said to be *red*. We call *boundary* of Y the set of points $x \in N(\delta)$ that are not red but are on the boundary of a cell $c \in C(\delta)$ containing a red point. We call *exterior* of Y the set of points $x \in N(\delta)$ that are neither in the boundary nor in the interior of Y. Points in the boundary (resp. exterior) of Y are said to be *blue* (resp. *green*). Notice that, for each cell $c \in C(\delta)$, $\tilde{c}$ cannot contain both red and green points. Also, given that $Y \neq \emptyset$, there is at least one red point, and given that G is connected and that $\mathsf{ground}(\delta) \setminus V_{\mathcal{S}^{\mathsf{g}}} \neq \emptyset$, there is at least one blue point.

For each face F of $\Sigma - \bigcup_{D \in \mathcal{D}} D$, let V_F be the set of blue points in the closure of F. Let Γ_H be a drawing without crossings in Σ whose vertices are the blue points and whose edges are, for each face F of $\Sigma - \bigcup_{D \in \mathcal{D}} D$ such that $|V_F| \geq 2$, edges between the blue points in V_F inducing a spanning tree. Notice that, given that Y is connected, Γ_H has at most one face containing red points, that we call the *red face* of Γ_H. The faces of Γ_H containing green points are called *green faces*.

Claim 5.2.34. *There is at most one connected component of Γ_H that bounds green faces of Γ_H.*

Proof of claim. Suppose not and let x and y be two blue points, g_x, g_y be two green points, and $\Delta_x, \Delta_y \in \mathcal{D}$ be disks such that x and y are not in the same connected component of Γ_H and that $x, g_x \in \mathsf{bd}(\Delta_x)$ and $y, g_y \in \mathsf{bd}(\Delta_y)$. Given that Γ_H is drawn in the projective plane Σ, at most one connected component of Γ_H may contain non-contractible cycles, and the other contains only contractible cycles. Hence, without loss of generality, we may assume that the connected component of x contains only (if any) contractible cycles. Then it is implied that there is a cycle T in $\bigcup_{\Delta \in \mathcal{D}} \mathsf{bd}(\Delta)$ avoiding blue points such that x and y are contained in different connected components of $\Sigma - T$ and such that the connected component of $\Sigma - T$ containing x is a disk. Given that the closure of a cell cannot contain both red and green points, it is implied that $T \cap N(\delta)$ contains either only red points, or only green points.

If $T \cap N(\delta)$ contains only red points, then g_x and g_y are in different components of $\Sigma - T$. This implies that $\bigcup_{S \in Y} \{v_S\}$ separates $\pi_\delta(g_x)$ from $\pi_\delta(g_y)$ in $\mathsf{torso}(G^\star_{\mathcal{S}}, \beta(r))$, and thus that

$N_{G^\star_S}(\bigcup_{S\in Y}\{v_S\}) \subseteq X$ separates $\pi_\delta(g_x)$ from $\pi_\delta(g_y)$ in G. This contradicts the fact that, for all $X' \subseteq X$, $G - X'$ is connected.

If $T \cap N(\delta)$ contains only green points, then by the definition of blue points, there is $\Delta'_x, \Delta'_y \in \mathcal{D}$ containing red points r_x and r_y respectively, such that $x \in \mathsf{bd}(\Delta'_x)$ and $y \in \mathsf{bd}(\Delta'_y)$, and Δ'_x and Δ'_y belong to different components of $\Sigma - T$. However, given that $\pi_\delta(r_x), \pi_\delta(r_y) \in Y$, this contradicts that fact that Y is connected. Hence the result. $\diamond$

Vortex containing V_Y. Let Γ_R be the connected component of Γ_H that bounds the green faces of Γ_H. If it does not exists, let Γ_R be any connected component of Γ_H. By definition of blue points, any edge of Γ_R must bound the red face. We now redefine the red face to be the face of Γ_R containing red points, that is, it contains the other components of Γ_H, if any. Let $x_1, \ldots, x_\ell$ be the blue points

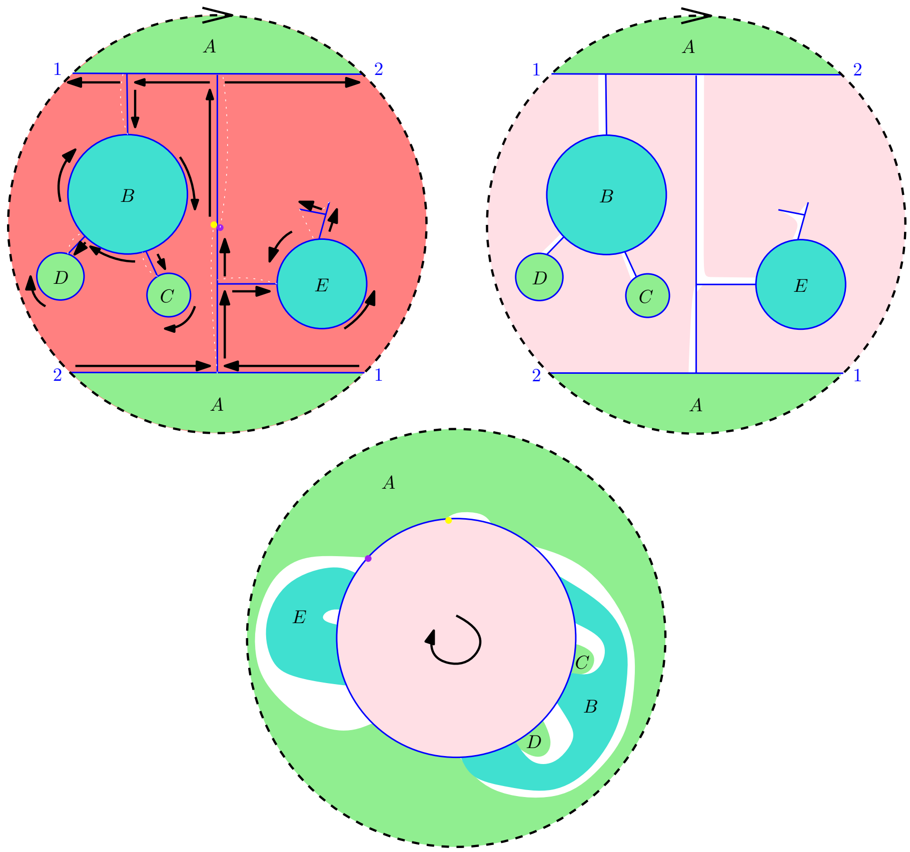

Figure 5.9: Illustration of Γ_R embedded on the projective plane: on the left, any point on the dashed cycle is identified to the point opposite to it with respect to the center of the cycle. From the yellow point, we add to the cyclic ordering the blue points that were not yet added following the order induced by the red face. The figure on the top right and bottom are the same up to some homeomorphism. The red face (figure on the left) becomes a vortex with boundary the blue points of Γ_R (figures on the right and bottom).

of Γ_R, in the order induced by their appearance on the boundary of the red face. Notice that, given that an edge may bound this face twice, some points may appear twice in the ordering. Hence, we only keep the first appearance of each point in the ordering, and this gives us a cyclic ordering Ω_Y of the blue points of Γ_R. See Figure 5.9 for an illustration.

Let C_Y be the set of all cells in the closure of the red face, and $\mathcal{D}_Y$ be the closure of the cells in

C_Y. Let G_Y be the graph obtained from $\bigcup_{c \in C_Y} \sigma_\delta(c)$ by removing the vertices that do not belong to $\beta(r)$. Let T be any simple closed curve drawn within the red face which contains all blue points of Γ_R, in the order prescribed by Ω_Y. If Σ is the sphere, then both components of $\Sigma - T$ are disks and if Σ is the projective plane, then exactly one component of $\Sigma - T$ is a disk (this would not be true if Σ was the torus or any other surface of higher Euler genus). Let c_Y be the (possibly unique) component of $\Sigma - T$ that is contained in the red face and D_Y be its closure. If c_Y is a disk, then let $\delta_Y = (\Gamma, \mathcal{D}_Y)$ be the Σ-decomposition of $G^\star_{\mathcal{S}}$ obtained from $\delta = (\Gamma, \mathcal{D})$ by setting $\mathcal{D}_Y = \mathcal{D} \setminus \mathcal{D}_Y \cup D_Y$. Hence, $C(\delta_Y) = C(\delta) \setminus C_Y \cup c_Y$, with $\sigma_{\delta_Y}(c_Y) = G_Y$, and $\pi_{\delta_Y}(\tilde{c}_Y) = \pi_\delta(V(\Gamma_R))$. If c_Y is not a disk, then we remove c_Y from Σ and replace it by a disk c'_Y, hence obtaining a sphere $\mathbb{S}_0$. Then, we let $\delta_Y = (\Gamma, \mathcal{D}_Y)$ be the $\mathbb{S}_0$-decomposition of $G^\star_{\mathcal{S}}$ obtained from δ by setting $\mathcal{D}_Y = \mathcal{D} \setminus \mathcal{D}_Y \cup D'_Y$, where D'_Y is the closure of c'_Y. Since any planar graph can be embedded in the projective plane, let us assume without loss of generality that δ_Y be the Σ-decomposition of $G^\star_{\mathcal{S}}$ with new vortex c_Y. Observe that every red point is now drawn in the interior of c_Y, and that no green point is drawn in c_Y. This implies that V_Y is drawn in the interior c_Y and that $V_{\mathcal{S}^{\mathsf{g}} \setminus Y} \subseteq \mathsf{ground}(\delta_Y) \setminus \pi_{\delta_Y}(\tilde{c}_Y)$.

We can apply this procedure iteratively to each connected component of $\mathcal{S}^{\mathsf{g}}$ (we do this procedure at most $18 f_{5.2.32}(k) + 1$ times by Claim 5.2.33). We thus obtain a Σ-decomposition δ' of $G^\star_{\mathcal{S}}$ such that, for each component Y of $\mathcal{S}^{\mathsf{g}}$, there is a cell $c_Y \in C(\delta')$ such that V_Y is drawn in the interior of c_Y. Hence, every stellation vertex is drawn in the interior of a cell of $C(\delta')$. It allows us to define a Σ-decomposition $\delta^\star$ of $\mathsf{torso}(G^\circ_{\mathcal{S}}, \beta(r))$ from the Σ-decomposition δ' of $G^\star_{\mathcal{S}}$ by keeping only the vertices of $\beta(r)$ and making a clique out of each Z^i_c for $c \in C(\delta)$ and out of each set $S \in \mathcal{S}$. Let C_0 be the union of the vortex cells of $\delta^\star$ and the cells of $\delta^\star$ whose interior is non empty. A vortex cell of $\delta^\star$ is either a vortex cell of δ or a cell c_Y for some component Y of $\mathcal{S}^{\mathsf{g}}$. Additionally, any cell of $\delta^\star$ whose interior is non empty is necessarily either a cell c_Y for some component Y of $\mathcal{S}^{\mathsf{g}}$ with $|\tilde{c}_Y| \leq 3$, or a cell containing a vertex of X_2. Therefore, $|C_0| \leq k - 1 + |\mathcal{S}^{\mathsf{g}}| + |X_2| \leq b_{5.2.32}(k)$.

For each component Y of $\mathcal{S}^{\mathsf{g}}$, let us construct a linear decomposition of width at most $w_{5.2.32}(k)$ of the vortex society (G_Y, Ω_Y) of c_Y. Remember that C_Y is the set of cells of δ that were replaced by c_Y. Let $A \subseteq (V(G_Y) \cap \mathsf{ground}(\delta)) \setminus V_Y$ be the set of non-stellation vertices of G_Y that are on the boundary of some non-vortex cell of C_Y in δ.

Claim 5.2.35. $|A| \leq |X_1| + 2|X_2|$.

Proof of claim. By definition, $A \cap X \subseteq X_1$. Let $v \in A \setminus X_1$. Then there is a non-vortex cell $c \in C_Y$ and $v_S \in V_Y$ such that $v, v_S \in \pi_\delta(\tilde{c})$. By the connectivity assumptions on δ, there is a path P from v_S to v whose internal vertices are in $\sigma_\delta(c) - \pi_\delta(\tilde{c})$. In particular, the neighbour of v_S in P is in X_2. The cell c contributes for at most two vertices of $A \setminus X_1$ (v and the third vertex of $\pi_\delta(\tilde{c})$, if it exists). Therefore, $|A \setminus X_1| \leq 2|X_2|$. ◇

Let $B \coloneqq (V(G_Y) \cap \mathsf{ground}(\delta)) \setminus A$. By definition of A and given that we assume any ground vertex of δ to be on the boundary of a non-vortex cell, it is implied that $B \subseteq V(\Omega_Y)$. Moreover, by our connectivity assumptions, each vertex of B is on the boundary of a (unique) vortex cell of C_Y. Let us fix a linear decomposition of each vortex cell of C_Y of adhesion at most $d_{5.2.26}(k)$. For each vertex v on the boundary of a (unique) vortex cell of C_Y, let Y_v be the bag corresponding to v, and let $Y'_v \coloneqq Y_v \cap \beta(r)$. We have $|Y'_v| \leq 2d_{5.2.26}(k) + 1$. Let $A' = V(G_Y) \setminus (\bigcup_{b \in B} Y'_b \setminus X)$. Let us bound the size of A'. Each vertex in A' is either a vertex of A, or a vertex of $X_2 \cup X_3$, or a vertex in $Y_a \setminus \{a\}$ for a vertex $a \in \mathsf{ground}(\delta)$ on the boundary of some non-vortex cell of C_Y, that is, for

$a \in A \cup V_Y$. Therefore,

$$\begin{aligned}|A'| &\leq |A| + |X_2| + |X_3| + (|A| + |\mathcal{S}^{\mathrm{g}}|) \cdot 2d_{5.2.26}(k)\\ &\leq |X_1| + 3|X_2| + |X_3| + (6|X_1| + 8|X_2| - 5) \cdot 2d_{5.2.26}(k)\\ &\leq (8|X| - 5) \cdot (2d_{5.2.26}(k) + 1).\end{aligned}$$

Then, we define a linear decomposition $(Z_v)_{v \in V(\Omega_Y)}$ of (G_Y, Ω_Y) as follows. For each $b \in B$, we set $Z_b \coloneqq Y_b' \cup A'$, and for each $a \in V(\Omega_Y) \setminus B$, we set $Z_a = A'$. $(Z_v)_{v \in V(\Omega_Y)}$ has width at most

$$\begin{aligned}|A'| + 2d_{5.2.26}(k) + 1 &\leq (8|X| - 4) \cdot (2d_{5.2.26}(k) + 1)\\ &\leq (24 f_{5.2.32}(k) + 4)(2d_{5.2.26}(k) + 1)\\ &\leq w_{5.2.32}(k).\end{aligned}$$

Therefore, $\delta^\star$ is an almost embedding of $\mathsf{torso}(G^{\circ}_{\mathcal{S}}, \beta(r))$ in Σ with vortex set C_0 of breadth at most $b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$. Hence, the result. $\square$

As corollary of Theorem 5.2.32, we immediately get the following.

Theorem 5.2.36. *Let $k \in \mathbb{N}$. Let G be a graph that excludes a long jump grid of order k as a minor. Then there exists a tree decomposition $\mathcal{T}$ of G of adhesion at most $3f_{5.2.32}(k) + 1$ such that the torso of $\mathcal{T}$ at each node has an almost embedding in the projective plane of breadth at most $b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$.*

Also, with the same proof as Theorem 5.2.32, but replacing Theorem 5.2.26 with Theorem 5.2.27, we conclude the following.

Theorem 5.2.37. *There exist functions $f_{5.2.37}, b_{5.2.37}, w_{5.2.37} : \mathbb{N}^2 \to \mathbb{N}$ such that the following holds. Let $k, c \in \mathbb{N}$. Let G be a graph that excludes a long jump grid of order k and a crosscap grid of order c as a minor. Then there exists a tree decomposition $\mathcal{T}$ of G of adhesion at most $3f_{5.2.37}(k,c) + 1$ such that the torso of $\mathcal{T}$ at each node has an almost embedding in the plane of breadth at most $b_{5.2.37}(k,c)$ and width at most $w_{5.2.37}(k,c)$.*

Moreover $f_{5.2.37}(k,c), b_{5.2.37}(k,c), w_{5.2.37}(k,c) = 2^{\mathcal{O}(k \log(k \cdot c))}$.

Identifying vortices

We now deduce our global structure theorem as it was stated in the introduction.

In [302], it was proved that vortices of bounded width have bounded bidimensionality.

Proposition 5.2.38 (Lemma 3.9, [302])**.** *For every graph G, every $k \in \mathbb{N}$, and every surface Σ with Euler genus at most g, if δ is a Σ-decomposition of G with width at most w and breadth at most b and $X = \bigcup\{\sigma(c) \mid c \text{ is a vortex of } \delta\}$, then $\mathsf{bg}(G, X) = \mathcal{O}((b^4 \cdot (b \cdot g \cdot w)^4)$.*

Therefore, we immediately get our upper bound.

Theorem 5.2.39. *Let $k \in \mathbb{N}$. Let G be a graph that excludes a long jump grid of order k as a minor. Then $\mathsf{idpr}^\star(G) = 2^{\mathcal{O}(k \log k)}$.*

Proof. By Theorem 5.2.36, there exists a tree decomposition $\mathcal{T} = (T, \beta)$ of G such that the torso G_t of $\mathcal{T}$ at each node $t \in V(T)$ has an almost embedding δ_t in the projective plane of breadth at most $b_{5.2.32}(k)$ and width at most $w_{5.2.32}(k)$. For $t \in V(T)$, let $\mathcal{C}_t$ be the set of all vortex cells of δ_t, $\mathcal{P}_t = (\sigma_{\delta_t}(c))_{c \in \mathcal{C}_t}$, and $X_t = \bigcup \mathcal{P}_t$. Given that $G_t /\!\!/ \mathcal{P}_t \in \mathcal{G}_{\mathsf{projective}}$, it implies that $\mathcal{I}(G_t, X_t) \cap \mathcal{G}_{\mathsf{projective}} \neq \emptyset$. The projective plane has Euler genus one. Hence, by Proposition 5.2.38, $\mathsf{bg}(G, X_t) = \mathcal{O}((b^4_{5.2.32} \cdot (b_{5.2.32} \cdot w_{5.2.32})^4) = 2^{\mathcal{O}(k \log k)}$. We conclude that $\mathsf{idpr}^\star(G) \leq \max_{t \in V(T)} \mathsf{bg}(G, X_t) = 2^{\mathcal{O}(k \log k)}$. $\square$

Similarly, we also conclude the following from Theorem 5.2.37 and Proposition 5.2.38.

Theorem 5.2.40. *Let $k, c \in \mathbb{N}$. Let G be a graph that excludes a long jump grid of order k and a crosscap grid of order c as a minor. Then* $\mathsf{idpl}^\star(G) = 2^{\mathcal{O}(k \log(k \cdot c))}$.

Combined with Proposition 5.1.1, Theorem 5.2.39 immediately implies Theorem 2.1.2. Additionally, we know from [136] that $\mathcal{G}_{\mathsf{projective}}$ is the set of all minors of crosscap grids $\mathcal{C}$. This, combined with Proposition 5.1.1 and Theorem 5.2.40, implies Theorem 2.1.4.

5.3 The lower bound

In this section, we prove Theorem 2.1.3. In fact we prove that $\mathsf{p}_{\mathscr{J}} \preceq \mathsf{idpr}^\star$ and Theorem 2.1.3 follows as all graphs in $\mathscr{J}$ are graphs in $\mathcal{G}_{\mathsf{edge\text{-}apex}}$.

Given a parameter p and a graph class $\mathcal{G}$, we define $\mathsf{p} \rhd_{\mathsf{id}} \mathcal{G} : \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ as the graph parameter where

$$\mathsf{p} \rhd_{\mathsf{id}} \mathcal{G}(G) \coloneqq \min\{k \mid \exists X \subseteq V(G), \mathsf{p}(G, X) \leq k \text{ and } \mathcal{I}(G, X) \cap \mathcal{G} \neq \emptyset\}.$$

Therefore, $\mathsf{idpr} = \mathsf{bg} \rhd_{\mathsf{id}} \mathcal{G}_{\mathsf{projective}}$. We prove the $\mathsf{p} \rhd_{\mathsf{id}} \mathcal{G}$ is a minor-monotone parameter.

Lemma 5.3.1. *Let p be a minor-monotone parameter and $\mathcal{G}$ be a minor-closed graph class. Then the parameter $p \rhd_{\mathsf{id}} \mathcal{G}$ is minor-monotone.*

Proof. Let $k = \mathsf{p} \rhd_{\mathsf{id}} \mathcal{G}(G)$. Then there is $X \subseteq V(G)$ and $\mathcal{P} = (X_1, \ldots, X_r) \in \mathcal{P}(X)$ be such that $p(G, X) = k$ and $G /\!\!/ \mathcal{P} \in \mathcal{G}$. Let H be a minor of G and $\{S_x \mid x \in V(H)\}$ be a model of H in G. For $i \in [r]$, let $Y_i = \{x \in V(H) \mid S_x \cap X_i \neq \emptyset\}$ and $Y = \bigcup_{i \in [r]} Y_i$. Given that p is minor-monotone, we have $\mathsf{p}(H, Y) \leq \mathsf{p}(G, X) = k$. Note that the sets Y_i may intersect, so $(Y_1, \ldots, Y_r)$ is not a partition. Let $\mathcal{P}' = (Y'_1, \ldots, Y'_s) \in \mathcal{P}(Y)$ be such that for each $i \in [r]$, there is $j \in [s']$ such that $Y_i \subseteq Y'_j$ and, if $Y'_j \setminus Y_i \neq \emptyset$, then there is $i' \in [r]$ such that $Y_{i'} \subseteq Y'_j$ and $Y_i \cap Y_{i'} \neq \emptyset$. Then, $\{S_x \mid x \in V(H) \setminus Y\} \cup \{\bigcup_{x \in Y'_j} S_x \setminus X \cup \{x_i \mid Y_i \subseteq Y'_j\} \mid j \in [s]\}$ is a model of $H /\!\!/ \mathcal{P}'$ in $G /\!\!/ \mathcal{P}$, where x_i is the vertex of $G /\!\!/ \mathcal{P}$ obtained from the identification of X_i. Given that $\mathcal{G}$ is minor-closed and that $G /\!\!/ \mathcal{P} \in \mathcal{G}$, it implies that $\mathcal{I}(H, Y) \neq \emptyset$. Therefore, $\mathsf{p} \rhd_{\mathsf{id}} \mathcal{G}(H) \leq k$. □

For the proof of Theorem 2.1.3, we first prove in Subsection 5.3.1 that $\mathsf{idpr}(\mathscr{J}_k) = \Omega(k^{1/4})$ (Lemma 5.3.3). Then, in Subsection 5.3.2, using the result of the previous subsection, we prove that $\mathsf{idpr}^\star(\mathscr{J}_k) = \Omega(k^{1/384})$ (Theorem 5.3.8). To do so, we need a result from [302] (Proposition 5.3.5) that essentially states that if the biggest grid parameter of each bag of a tree decomposition of G is small, then so is the biggest grid parameter of G.

By Lemma 5.3.1, $\mathsf{idpr} = \mathsf{bg} \rhd_{\mathsf{id}} \mathcal{G}_{\mathsf{projective}}$ is minor-monotone, and thus so is $\mathsf{idpr}^\star$. Therefore, for any graph G containing $\mathscr{J}_k$ as a minor, $\mathsf{idpr}^\star(G) = \Omega(k^{1/384})$, and therefore, $\mathsf{p}_{\mathscr{J}} \preceq \mathsf{idpr}^\star$.

5.3.1 Identifications in a long-jump grid

We first prove that $\mathsf{p}_{\mathscr{J}} \preceq \mathsf{idpr}$.

We define $\mathscr{J}' = \{\mathscr{J}'_k \mid k \in \mathbb{N}\}$ where $\mathscr{J}'_k$ is the graph obtained from a $(4k+1) \times (4k+1)$-grid Γ_{4k+1} after adding the edge $\{x, y\} = \{(k+1, 2k+1), (3k+1, 2k+1)\}$ (see Figure 5.10 for an example where $k = 8$). It is easy to see that $\mathsf{p}_{\mathscr{J}}$ and $\mathsf{p}_{\mathscr{J}'}$ are linearly equivalent.

Let F be the graph depicted in Figure 5.10 (on the right). F is known to be a minor-obstruction of the projective plane (see $\mathcal{D}_{17}$ in [19, 233]).

We need the following result.

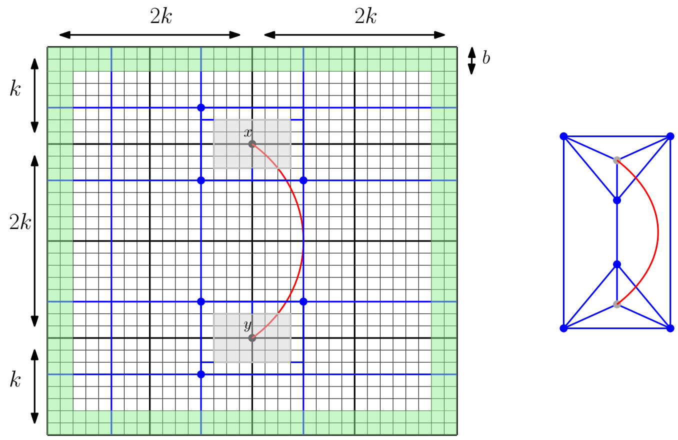

Figure 5.10: The graph $\mathscr{J}'_8$ and a non-projective graph F.

Proposition 5.3.2 ([82]). *Let Γ_m be the $(m \times m)$-grid and S be a subset of the vertices in the central $(m-2\ell) \times (m-2\ell)$-subgrid of Γ_m, where $\ell = \lfloor \sqrt[4]{|S|} \rfloor$. Then G contains the $(\ell \times \ell)$-grid as an S-minor.*

We now have all ingredients for the main result of this part. We essentially prove that if a set $X \subseteq V(\mathscr{J}'_k)$ has too small biggest grid parameter, then every graph in $\mathcal{I}(\mathscr{J}'_k, X)$ contains F as a minor. And since F is not projective, it thus implies that $\mathcal{I}(\mathscr{J}'_k, X) \cap \mathcal{G}_{\mathsf{projective}} = \emptyset$.

Lemma 5.3.3. *Let $k \in \mathbb{N}$. Then $\mathsf{idpr}(\mathscr{J}_k) = \Omega(k^{1/4})$.*

Proof. Given that $\mathsf{p}_{\mathscr{J}}$ and $\mathsf{p}_{\mathscr{J}'}$ are linearly equivalent, it is enough to prove the result for $\mathsf{p}_{\mathscr{J}'}$. Let $X \subseteq V(\mathscr{J}'_k)$ and $\mathcal{P} \in \mathcal{P}(X)$ be such that $G' := \mathscr{J}'_k /\!\!/ \mathcal{P} \in \mathcal{G}_{\mathsf{projective}}$. Let us prove that $\mathsf{bg}(\mathscr{J}'_k, X) = \Omega(k^{1/4})$.

For each margin $b \in [0, \lfloor k/2 \rfloor]$, let X_b be the set of vertices of X that belong in the $(k-b) \times (k-b)$-subgrid Γ_{k-b} of Γ_{4k+1} and $\mathcal{P}_b = \mathcal{P} \cap X_b$. We claim that $|X_b| \geq k-b$. Suppose towards a contradiction that $|X_b| \leq k-b-1$. Then, in Γ_{k-b}, there are at least two vertical paths on the left of x and y, one vertical path on the right of x and y, one horizontal path above x, two horizontal paths between x and y, and one horizontal path below y, whose vertices do not intersect X (drawn in blue in Figure 5.10). Let $V_x, V_y \subseteq V(\Gamma_{k-b})$ be the sets of all the vertices in the two squares containing x and y surrounded by the aforementioned paths (drawn in grey in Figure 5.10), respectively. Let $\mathcal{P}'$ be obtained from $\mathcal{P}$ by adding two new parts: V_x and V_y. Given that $G[V_x]$ (resp. $G[V_y]$) is connected, the identification of all vertices in V_x (resp. V_y) is equivalent to the contraction of all edges in $G[V_x]$ (resp. $G[V_y]$). Given that $\mathscr{J}'_k /\!\!/ \mathcal{P} \in \mathcal{G}_{\mathsf{projective}}$ and that $\mathcal{G}_{\mathsf{projective}}$ is closed under contractions, we thus have $G' /\!\!/ V_x /\!\!/ V_y = G /\!\!/ \mathcal{P}' \in \mathcal{G}_{\mathsf{projective}}$. However, as depicted in Figure 5.10, F is a minor of $G /\!\!/ \mathcal{P}'$, a contradiction. We conclude that $|X_b| \geq k-b$.

We choose the margin b_k as the maximum integer b such that $\lfloor \sqrt[4]{k-b} \rfloor \geq 2b$. This implies that $\lfloor \sqrt[4]{|X_{b_k}|} \rfloor \geq 2b_k$, therefore, from Proposition 5.3.2, $\mathsf{bg}(\mathscr{J}'_k, X) \geq \lfloor \sqrt[4]{|X_{b_k}|} \rfloor = 2b_k$. As $b_k = \Omega(k^{1/4})$, we are done. □

5.3.2 Lower bound under the presence of clique-sums

In this section we extend the polynomial bound in Lemma 5.3.3 from idpr to $\mathsf{idpr}^\star$. For this we need a series of definitions and preliminary results from [302].

We say that a graph G is (f,k)-*tightly connected* for some $f:\mathbb{N}\to\mathbb{N}$ and $k\in\mathbb{N}$, if for every separation (B_1,B_2) of G of order $q<k$ such that both $G[B_1\setminus B_2]$ and $G[B_2\setminus B_1]$ are connected, it holds that one of B_1, B_2 has at most $f(q)$ vertices. We say that a parametric graph $\mathscr{H}=\langle\mathscr{H}_k\rangle_{k\in\mathbb{N}}$ is f-*tightly connected* if for each $k\in\mathbb{N}$, $\mathscr{H}_k$ is (f,k)-tightly connected.

Proposition 5.3.4 (Lemma 4.9, [302]). *Let $r\in\mathbb{N}$ and let $g,h:\mathbb{N}\to\mathbb{N}$ be two non-decreasing functions. Let p be a minor-monotone graph parameter such that for every graph H, $\mathsf{hw}(H)\le h(\mathsf{p}(H))$. Let G be a $(g,h(r)+1)$-tightly connected graph with $|V(G)|>2g(h(r))$ and $\mathsf{p}^\star(G)\le r$. Then G admits a tree decomposition (T,β) where T is a star with center t, where for the torso G_t of (T,β) at t, $\mathsf{p}(G_t)\le r$, and where, for every $e=tt'\in E(T)$, $G[\beta(t')\setminus\beta(t)]$ is a connected graph and $|\beta(t')|\le g(h(r))$.*

Proposition 5.3.5 (Lemma 4.11, [302]). *Let G be a 4-connected graph of maximum degree Δ. Let $r\in\mathbb{N}$ and let (T,β) be a tree decomposition of G where T is a star with center t and such that for every $e=tt'\in E(T)$, $G[\beta(t')\setminus\beta(t)]$ is a connected graph on at most l vertices. Let G_t be the torso of (T,β) at t and we denote $G^c=(V(G),E(G)\cup E(G_t))$, $m=\mathsf{hw}(G^c)$, and $B=\bigcup_{t'\in V(T)\setminus\{t\}}\beta(t')$. There is a function $f_{5.3.5}:\mathbb{N}^4\to\mathbb{N}$ such that if X is a subset of $V(G_t)$ where $\mathsf{bg}(G_t,X)\le r$, then $\mathsf{bg}(G^c,X\cup B)\le f_{5.3.5}(r,m,l,\Delta)$. Moreover, $f_{5.3.5}(r,m,l,\Delta)\in\mathcal{O}(m^{48}+r^{1/2}\cdot(m^8l^4+\Delta m^{24}l^2))\subseteq\mathsf{poly}(m+l+r+\Delta)$.*

Lemma 5.3.6. *Let $\mathcal{G}$ be a minor-closed graph class of Hadwiger number at most η and G be a graph. Then $\mathsf{hw}(G)\le\mathsf{bg}\triangleright_{\mathsf{id}}\mathcal{G}(G)^2+\eta$.*

Proof. Let $t=\mathsf{hw}(G)$ and $k=\mathsf{bg}\triangleright_{\mathsf{id}}\mathcal{G}(G)$. Given that bg is minor-monotone and that $\mathcal{G}$ is minor-closed, by Lemma 5.3.1, $\mathsf{bg}\triangleright_{\mathsf{id}}\mathcal{G}$ is minor-monotone. Therefore, given that K_t is a minor of G, there exists $Y\subseteq V(K_t)$ such that $\mathsf{bg}(K_t,Y)\le k$ and $\mathcal{I}(K_t,Y)\cap\mathcal{G}\neq\emptyset$. Let $G'\in\mathcal{I}(K_t,Y)\cap\mathcal{G}$. Given that $\mathsf{hw}(G')\le\eta$, the identification of a partition of Y reduced the size by at least $t-\eta$. Hence, given that identifying $p+1$ vertices reduce that size by at most p, we conclude that $|Y|\ge t-\eta+1$. Notice that $k\ge\mathsf{bg}(K_t,Y)\ge\mathsf{bg}(K_t[Y])\ge\lfloor|Y|^{1/2}\rfloor\ge(t-\eta)^{1/2}$. This implies that $t\le k^2+\eta$. □

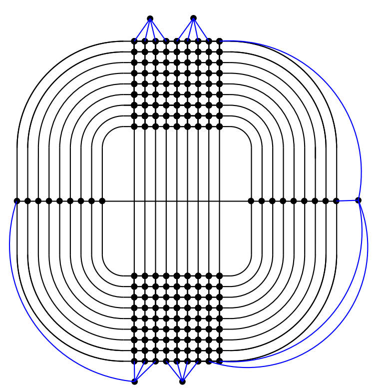

Figure 5.11: The graph $\mathscr{I}_4$.

A new variant of the long-jump grid. For the purposes of the proofs in this section, we consider an "enhanced version" of $\mathscr{J}_k$, namely $\mathscr{I}=\langle\mathscr{I}_k\rangle_{k\in\mathbb{N}}$ which is obtained from $\mathscr{J}_{2k+1}$ as follows. We partition the $4k+4$ vertices in the outermost cycle of $\mathscr{J}_{2k+1}$ into $k+1$ sets of four consecutive vertices. For each such set, we add a new vertex adjacent to the four vertices of the set, as illustrated in Figure 5.11. These $k+1$ new vertices are called *satellite* vertices. Note that $\mathscr{I}_k$ is 4-connected and 4-regular. Moreover, we easily have that $\mathsf{p}_{\mathscr{J}}$ and $\mathsf{p}_{\mathscr{I}}$ are linearly equivalent.

Lemma 5.3.7. *$\mathscr{J}$ is $f_{5.3.7}$-tightly connected for $f_{5.3.7}(q) = (2q+1)^2$.*

Proof. Let $G := \mathscr{I}_k$. Let (B_1, B_2) be a separation of G of order $q < k$. Let $A = B_1 \cap B_2$, $D_1 = B_1 \setminus A$, and $D_2 = B_2 \setminus A$. We assume that both $G[D_1]$ and $G[D_2]$ are connected. Let us show that one of B_1 and B_2 has at most $(2q+1)^2$ vertices.

Let $k' = 2k+1$. Let Q be the spanning $((2k'+2) \times k')$-annulus grid of G. Clearly Q contains k' cycles. Also it contains $2k'+2$ paths, each on k' vertices, which we call *tracks*. We also use R for the set of the satellite vertices of $\mathscr{I}_k$. We enhance the tracks by extending each of them to the unique satellite vertex that is adjacent to one of its endpoints. Let Y be the union of all the $\geq k'-q$ cycles that are *not* met by A and of all $\geq 2k'+2-4q$ tracks that are *not* met by A (observe that if a vertex of A belongs to R, then it meets four tracks).

As every track in Y has a common endpoint to every cycle in Y, we obtain that Y is connected, therefore $V(Y)$ is either a subset of D_1 or a subset of D_2. W.l.o.g., we assume that $V(Y) \subseteq D_1$. We next prove that $|B_2| \leq (2q+1)^2$.

Let x be a vertex of D_2 and let P_x be some path of $G[B_2]$ starting from x and finishing to some vertex of A and with all internal vertices in D_2. This path cannot meet more than $q = |A|$ cycles of Q because each such cycle contains some vertex of $Y \subseteq D_1$. Similarly, P_x cannot meet more than $q = |A|$ tracks as each such track contains some vertex of $Y \subseteq D_1$. This implies that every vertex of $G[B_2]$ should be within distance at most q from x. It is now easy to verify that, in G, the vertices within distance at most q from some $x \in A$ is upper bounded by $(2q+1)^2$. As all vertices of B_2 are accessible from x within this distance in $G[B_2]$, we conclude that $|B_2| \leq (2q+1)^2$. □

Theorem 5.3.8. *Let $k \in \mathbb{N}$. Then $\mathsf{idpr}^\star(\mathscr{J}_k) = \Omega(k^{1/384})$.*

Proof. We prefer to work with $\mathscr{I}_k$ (cf. Figure 5.11) because it is 4-connected and 4-regular. By Lemma 5.3.3, we obtain that $\mathsf{idpr}(\mathscr{J}_k) = \Omega(k^{1/4})$. Thus, given that $\mathscr{J}_k$ and $\mathscr{I}_k$ are linearly equivalent, it is enough to prove that

$$\mathsf{idpr}(\mathscr{I}_k) \in \mathcal{O}\big(\mathsf{idpr}^\star(\mathscr{I}_k)^{96}\big).$$

For simplicity, we set $G = \mathscr{I}_k$. Let r be such that $\mathsf{idpr}^\star(G) \leq r$. Our objective is to prove that $\mathsf{idpr}(G) \in \mathcal{O}(r^{96})$.

Let $\eta = \max\{\mathsf{hw}(G) \mid G \text{ is projective}\} = 6$. Let $h, g : \mathbb{N} \to \mathbb{N}$ be such that

$$\begin{aligned} h(x) &= x^2 + \eta, \text{ and} && (5.1)\\ g(x) &= (2x+1)^2. && (5.2) \end{aligned}$$

Thus, by applying Lemma 5.3.6 with $\mathcal{G} = \mathcal{G}_{\mathsf{projective}}$, for any graph H, we have

$$\mathsf{hw}(H) \leq h(\mathsf{idpr}(H)). \tag{5.3}$$

Given that $|V(G)| = 8k^2 + 13k + 5$, and that $\mathsf{idpr}(G) \leq |V(G)|$, we may assume that

$$\begin{aligned} |V(G)| &> 2g(h(r)), \text{ and} && (5.4)\\ h(r)+1 &\leq k && (5.5) \end{aligned}$$

Indeed, otherwise, in the first case, $\mathsf{idpr}(G) \leq 2g(h(r)) \in \mathcal{O}(r^4) \subseteq \mathcal{O}(r^{96})$, and in the second case, $\mathsf{idpr}(G) \leq 8k^2 + 13k + 5 \leq 8h(r)^2 + 13h(r) + 5 \in \mathcal{O}(r^4) \subseteq \mathcal{O}(r^{96})$. In both cases, we are done. By

Lemma 5.3.7, G is (g, k)-tightly connected, and thus, by (5.5), G is $(g, h(r)+1)$-tightly connected. Thus, we have all ingredients to apply Proposition 5.3.4. We obtain a tree decomposition (T, β) of G where T is a star with center t, where $\mathsf{idpr}(G_t) \leq r$ and where for every $tt' \in E(T')$, $G[\beta(t') \setminus \beta(t)]$ is a connected graph and $|\beta(t')| \leq g(h(r))$.

Since $\mathsf{idpr}(G_t) \leq r$, there is a set $X_t \subseteq \beta(t)$ and $\mathcal{P}_t = (X_1, \ldots, X_p) \in \mathcal{P}(X_t)$ such that $G_t /\!\!/ \mathcal{P}_t \in \mathcal{G}_{\mathsf{projective}}$ and $\mathsf{bg}(G_t, X_t) \leq r$. Let G^c denote the graph $(V(G), E(G) \cup E(G_t))$ and $B = \bigcup_{t' \in V(T) \setminus \{t\}} \beta(t')$. By (5.3), $\mathsf{hw}(G^c) \leq h(r)$. Given that G is 4-connected and has maximum degree four, we can hence apply Proposition 5.3.5 with $\Delta = 4, l = g(h(r))$ and $m = h(r)$ on the tree decomposition (T, β). We conclude that

$$\mathsf{bg}(G^c, X_t \cup B) \quad \leq \quad f_{5.3.5}(r, \eta, g(h(r)), 4) \in \mathcal{O}(r^{96}). \tag{5.6}$$

Clearly G is a (spanning) subgraph of G^{c}. For each $t' \in N'_T(t)$, let $A_{t'}$ be the adhesion of t' and t, let $C_{t'} = \beta(t') \setminus A_{t'}$, and let $Y_{t'}$ be a set composed of $C_{t'}$ and an arbitrary vertex $a_{t'}$ of $A_{t'}$. Remember that $G^c[A_{t'}]$ is a clique. Hence, the identification of $C_{t'}$ with $a_{t'}$ does not create additional edges, and thus has the same effect as removing $C_{t'}$. Therefore, $G_t = G^c /\!\!/ (Y_{t'})_{t' \in N_T(t)}$, and thus $G^c /\!\!/ (Y_{t'})_{t' \in N_T(t)} /\!\!/ \mathcal{P}_t \in \mathcal{G}_{\mathsf{projective}}$. Thus $\mathsf{idpr}(G^c) \leq \mathsf{bg}(G^{\mathsf{c}}, X_t \cup B_t) \in^{(5.6)} \mathcal{O}(r^{96})$. Given that $\mathsf{idpr}(G) \leq \mathsf{idpr}(G^c)$, the statement follows. □

If we want to go to the plane instead of the projective plane, the following lemma is an easy corollary of Theorem 5.3.8 and [302, Lemma 7.13]. Remember that $\mathscr{C}_k$ is the crosscap grid.

Theorem 5.3.9. *Let $k \in \mathbb{N}$. Then $(\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{planar}})^\star(\mathscr{J}_k) = \Omega(k^{1/384})$ and $(\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{planar}})^\star(\mathscr{C}_k) = \Omega(k^{1/480})$.*

Proof. By Theorem 5.3.8, $(\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{projective}})^\star(\mathscr{J}_k) = \Omega(k^{1/384})$. Given that $\mathcal{G}_{\mathsf{planar}} \subseteq \mathcal{G}_{\mathsf{projective}}$, we thus have $\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{planar}} \geq \mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{projective}}$, and therefore $(\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{planar}})^\star(\mathscr{J}_k) = \Omega(k^{1/384})$.

By [302, Lemma 7.13], $(\mathsf{bg} \triangleright_{\mathsf{rem}} \mathcal{G}_{\mathsf{planar}})^\star(\mathscr{C}_k) = \Omega(k^{1/480})$, where $\mathsf{bg} \triangleright_{\mathsf{rem}} \mathcal{G}_{\mathsf{planar}} : \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$ is the graph parameter where

$$\mathsf{bg} \triangleright_{\mathsf{rem}} \mathcal{G}_{\mathsf{planar}}(G) \coloneqq \min\{k \mid \exists X \subseteq V(G), \mathsf{bg}(G, X) \leq k \text{ and } G - X \in \mathcal{G}_{\mathsf{planar}}\}.$$

Given that $\mathcal{I}(G, X) \in \mathcal{G}_{\mathsf{planar}}$ implies $G - X \in \mathcal{G}_{\mathsf{planar}}$, we thus have $\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{planar}} \geq \mathsf{bg} \triangleright_{\mathsf{rem}} \mathcal{G}_{\mathsf{planar}}$, and therefore $(\mathsf{bg} \triangleright_{\mathsf{id}} \mathcal{G}_{\mathsf{planar}})^\star(\mathscr{C}_k) = \Omega(k^{1/480})$. □

Part III

Towards efficiency

CHAPTER 6

Identification to forests

Contents

In this chapter, we prove the results presented in Section 2.2, which are restated here for convenience.

Theorem 2.2.1. IDENTIFICATION TO FOREST *is* NP*-complete.*

Theorem 2.2.2. *There is an algorithm that, given an instance (G, k) of* IDENTIFICATION TO FOREST, *outputs in time $\mathcal{O}(k\sqrt{\log k} \cdot n + k^3)$ an equivalent instance (G', k') where $|V(G')| \leq 2k+1$ and $k' \leq k+1$. Alternatively, one can solve* IDENTIFICATION TO FOREST *in time $\mathcal{O}(1.2738^k + k\sqrt{\log k} \cdot n)$.*

Theorem 2.2.3. *Let $k \in \mathbb{N}$. The obstructions of $\mathcal{F}^{(k)}$ have at most $2k+4$ vertices.*

In particular, Theorem 2.2.1 and Theorem 2.2.2 are proved in Section 6.1 and Section 6.2 is dedicated to Theorem 2.2.3. Additionally, "universal obstructions" for IDENTIFICATION TO FOREST are given in Section 6.3, the relationship between identification and contraction is detailed in Section 6.4, and we introduce "identification minors" in Section 6.5 and discuss the link between WQO and the identification operation.

Some notations and reminders. Let us first give a few additional notations for this chapter. The class of forests is denoted by $\mathcal{F}$. Let $\mathcal{H}$ be a graph class and G be a graph. We say that a partition $\mathcal{X} \in \mathcal{P}(G)$ is an *id-$\mathcal{H}$ partition of G* if $G /\!\!/ \mathcal{X} \in \mathcal{H}$. The *order* of a partition $\mathcal{X} \in \mathcal{P}(G)$ is $|\cup \mathcal{X}|$. A *minimum* id-$\mathcal{H}$ partition of G is an id-$\mathcal{H}$ partition of G of minimum order and this minimum order is denoted by $\mathsf{idf}(G)$ when $\mathcal{H} = \mathcal{F}$. Similarly, the minimum size of a vertex cover of G is denoted by $\mathsf{vc}(G)$. As explained in the introduction, the problem of Identification to $\mathcal{H}$ asks, given a graph G and a non-negative integer k, whether G admits an id-$\mathcal{H}$ partition of order at most k. Finally, we denote by $\mathcal{H}^{(k)}$ the set of graphs that admit an id-$\mathcal{H}$ partition of order k.

6.1 Hardness and parameterized results

In this section we exploit the relation between Identification to Forest and Vertex Cover (cf. Subsection 6.1.1) to present a hardness result (cf. Subsection 6.1.2), a linear kernel, and an FPT-algorithm (cf. Subsection 6.1.3) for Identification to Forest, building on the corresponding results for Vertex Cover.

6.1.1 Dealing with bridges

In this part, we prove that Identification to Forest can be reduced to Vertex Cover (Lemma 6.1.5).

We first present two observations concerning identifications that imply that we can consider each connected component of a graph separately.

Observation 6.1.1. *Let $\mathcal{H}$ be a hereditary graph class and G be a graph. Then, for every $\mathcal{X} \in \mathcal{P}(G)$, if $G /\!\!/ \mathcal{X} \in \mathcal{H}$, then for each $H \in \mathsf{cc}(G)$, $H /\!\!/ (\mathcal{X} \cap V(H)) \in \mathcal{H}$.*

Proof. Let $H \in \mathsf{cc}(G)$. Given that $H /\!\!/ (\mathcal{X} \cap V(H)) = G /\!\!/ \mathcal{X} - (V(G) \setminus V(H))$ and that $\mathcal{H}$ is hereditary, we conclude that $H /\!\!/ (\mathcal{X} \cap V(H)) \in \mathcal{H}$. $\square$

Observation 6.1.2. *Let $\mathcal{H}$ be a graph class that is closed under disjoint union and G be a graph. Then, for each $H \in \mathsf{cc}(G)$ and for each $\mathcal{X}_H \in \mathcal{P}(H)$, if $H /\!\!/ \mathcal{X}_H \in \mathcal{H}$, then $G /\!\!/ \bigcup_{H \in \mathsf{cc}(G)} \mathcal{X}_H \in \mathcal{H}$.*

Proof. Given that $\mathcal{H}$ is closed under disjoint union and $G /\!\!/ \bigcup_{H \in \mathsf{cc}(G)} = \bigcup_{H \in \mathsf{cc}(G)} H /\!\!/ \mathcal{X}_H$, we conclude that $G /\!\!/ \bigcup_{H \in \mathsf{cc}(G)} \mathcal{X}_H \in \mathcal{H}$. $\square$

We now prove that we can safely delete bridges.

Lemma 6.1.3. *Let G be a graph and G^{b} be the graph obtained from G after removing all bridges. Then $\mathsf{idf}(G) = \mathsf{idf}(G^{\mathsf{b}})$.*

Proof. Let $k := \mathsf{idf}(G)$. By definition, $G \in \mathcal{F}^{(k)}$. By Lemma 5.3.1 (with $\mathsf{p} = \mathsf{size}$), $\mathcal{F}^{(k)}$ is minor-closed, so $G - e \in \mathcal{F}^{(k)}$ for any edge e of G. Therefore, $\mathsf{idf}(G - e) \leq \mathsf{idf}(G)$.

By Observation 6.1.1 and Observation 6.1.2, we may assume without loss of generality that G is connected. Let e be a bridge of G. Let G_1 and G_2 be the two connected components of $G - e$. For $i \in [2]$, let $\mathcal{X}_i \in \mathcal{P}(X_i)$ be a minimum id-$\mathcal{F}$ partition of G_i. By Observation 6.1.2, $(G - e) /\!\!/ \mathcal{X} \in \mathcal{F}$ where $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$. Suppose towards a contradiction that $G /\!\!/ \mathcal{X}$ contains a cycle C. Then, given that $(G - e) /\!\!/ \mathcal{X}$ is acyclic, it implies that e is an edge of C. Given that no part of $\mathcal{X}$ contains vertices of both G_1 and G_2, it implies that e is already an edge of a cycle. This contradicts the fact that e is a bridge. The lemma follows by repeatedly applying this argument as long as there is a bridge. $\square$

We can then prove that $\mathsf{idf} = \mathsf{vc}$ on bridgeless graphs.

Lemma 6.1.4. *Let G be a bridgeless graph. Then* $\mathsf{idf}(G) = \mathsf{vc}(G)$.

Proof. Let X be a vertex cover of G. Then $G /\!\!/ X$ is a star (if G is edgeless, a vertex is considered as a star). Hence, $(X) \in \mathcal{P}(G)$ is an id-$\mathcal{F}$ partition of G. So $\mathsf{idf}(G) \leq \mathsf{vc}(G)$.

Let $\mathcal{X}$ be an id-$\mathcal{F}$ partition of G. Let $F := G /\!\!/ \mathcal{X} \in \mathcal{F}$. Let us color red the vertices of F that are heirs of a part of $\mathcal{X}$, and blue the other vertices of F. Given that G is bridgeless, F contains no edge whose endpoints are both blue. Hence, the red vertices form a vertex cover of F. Therefore, X is a vertex cover of G. So $\mathsf{vc}(G) \leq \mathsf{idf}(G)$. □

Finally, we get the main result of this part as a direct corollary of Lemma 6.1.3 and Lemma 6.1.4.

Lemma 6.1.5. *Let G be a graph and G^{b} be the graph obtained from G after removing all bridges. Then* $\mathsf{idf}(G) = \mathsf{vc}(G^{\mathsf{b}})$.

6.1.2 NP-completeness

In this part, we prove Theorem 2.2.1 (more precisely Theorem 6.1.7).

Before proving the NP-completeness of Identification to Forest, we first need the following lemma.

Lemma 6.1.6. *Let $k \in \mathbb{N}$. Let G be a graph. Then there is a bridgeless graph G' with $|V(G)| + 1$ vertices such that $G \in \mathcal{V}_k$ if and only if $G' \in \mathcal{V}_{k+1}$. Moreover, G' can be constructed in linear time.*

Proof. Let us construct a bridgeless graph G' from G. Let I be the set of isolated vertices of G. We add a new vertex v to G and add an edge between v and every vertex of $G - I$. The constructed graph G' is clearly bridgeless.

Let us check that $G \in \mathcal{V}_k$ if and only if $G' \in \mathcal{V}_{k+1}$. We assume that G has at least one edge, otherwise the claim is trivially true. Suppose the $G \in \mathcal{V}_k$ and let X be a vertex cover of G of size at most k. Then $X \cup \{v\}$ is a vertex cover of G so $G' \in \mathcal{V}_{k+1}$. Suppose now that $G' \in \mathcal{V}_{k+1}$. Let Y be a vertex cover of G' of size at most $k + 1$. If $v \in Y$, then $Y \setminus \{v\}$ is a vertex cover of G of size at most k. Otherwise $v \notin Y$. It implies that $V(G) \setminus I \subseteq Y$. But then, for any vertex x of $G - I$, $N_{G-I}(x) \subseteq Y$. Therefore, $Y \setminus \{x\}$ is a vertex cover of G of size at most k. Hence, $G \in \mathcal{V}_k$.

Given that G' can be constructed in linear time, the result follows. □

Theorem 6.1.7. Identification to Forest *is* NP*-complete.*

Proof. Given a graph G and a partition $\mathcal{X} \in \mathcal{P}(G)$, checking that $G /\!\!/ \mathcal{X} \in \mathcal{F}$ can obviously be done in linear time. We reduce from Vertex Cover that is NP-hard [181]. Let G be a graph. Let b be the number of bridges in G. If $b \geq 1$, by Lemma 6.1.6, there is a graph G' with $|V(G)| + 1$ vertices such that $G \in \mathcal{V}_k$ if and only if $G' \in \mathcal{V}_{k'}$ where $k' := k + 1$. If $b = 0$, we set $G' := G$ and $k' := k$. Since $G' \in \mathcal{V}_{k'}$ is bridgeless, by Lemma 6.1.4, $G' \in \mathcal{F}^{(k')}$. Since G' can be constructed in linear time and that $G \in \mathcal{V}_k$ if and only if $G' \in \mathcal{F}^{(k')}$, the result follows. □

6.1.3 Parameterized results for Identification to Forest

In this part we prove Theorem 2.2.2 (alternatively Theorem 6.1.10).

The following kernelization result is known for Vertex Cover.

Proposition 6.1.8 ([58])**.** *Given an instance (G, k) of* Vertex Cover*, one can compute in time $\mathcal{O}(kn + k^3)$ an equivalent instance (G', k') such that $k' \leq k$ and $|V(G')| \leq 2k' \leq 2k$.*

Additionally, the current best FPT-algorithm for Vertex Cover is the following.

Proposition 6.1.9 ([59])**.** *There is an algorithm solving* Vertex Cover *in time* $\mathcal{O}(1.2738^k + kn)$*.*

Hence, we can derive the following kernalization and FPT results for Identification to Forest.

Theorem 6.1.10. *Given an instance* (G, k) *of* Identification to Forest*, one can compute in time* $\mathcal{O}(k\sqrt{\log k} \cdot n + k^3)$ *an equivalent instance* (G', k') *such that* $|V(G')| \leq 2k + 1$ *and* $k' \leq k + 1$*. Alternatively, one can solve* Identification to Forest *in time* $\mathcal{O}(1.2738^k + k\sqrt{\log k} \cdot n)$*.*

Proof. By Lemma 5.3.1 (with $\mathsf{p} = \mathsf{size}$) and Proposition 4.2.1, we may assume that $m := |E(G)| = \mathcal{O}(k\sqrt{\log k} \cdot n)$, given that (K_{3+k}, k) is a no-instance of Identification to Forest.

Let G_1 be obtained from G after removing bridges (which can be done in time $\mathcal{O}(n + m)$). By Lemma 6.1.5, (G, k) is a yes-instance of Identification to Forest if and only if (G_1, k) is a yes-instance of Vertex Cover. Hence, for the FPT result, it remains to solve Vertex Cover in time $\mathcal{O}(1.2738^k + kn)$ using Proposition 6.1.9.

For the kernelization result, it goes as follows. By Proposition 6.1.8 and the above discussion, we can construct in time $\mathcal{O}(kn + k^3)$ an instance (G_2, k_2) with $|V(G_2)| \leq 2k_2 \leq 2k$ such that (G_1, k) is a yes-instance of Vertex Cover if and only if (G_2, k_2) is a yes-instance of Vertex Cover. By Lemma 6.1.6, we can construct in linear time an instance (G_3, k_3) such that G_3 is a bridgeless graph with $|V(G_3)| \leq |V(G_2)| + 1$ and $k_3 \leq k_2 + 1$ such that (G_2, k_2) is a yes-instance of Vertex Cover if and only if (G_3, k_3) is a yes-instance of Vertex Cover. Finally, by Lemma 6.1.5, given that G_3 is bridgeless, by Lemma 6.1.4, (G_3, k_3) is a yes-instance of Vertex Cover if and only if it is a yes-instance of Identification to Forest. Hence the result. ☐

6.2 Obstructions

Given that Vertex Cover and Identification to Forest are strongly related, it is reasonable to suspect that this holds for their obstructions as well. Already, as a direct corollary of Lemma 6.1.5, we have the two following results.

Observation 6.2.1. *Let* $k \in \mathbb{N}$ *and* $F \in \mathsf{obs}(\mathcal{F}^{(k)})$*. Then* F *is bridgeless.*

Lemma 6.2.2. *Let* $k \in \mathbb{N}$*. The bridgeless obstructions of* $\mathcal{V}_k$ *are obstructions of* $\mathcal{F}^{(k)}$*.*

Proof. Let $H \in \mathsf{obs}(\mathcal{V}_k)$ be bridgeless. By Lemma 6.1.4, $H \notin \mathcal{F}^{(k)}$. Thus, there is a minor H' of H such that $H' \in \mathsf{obs}(\mathcal{F}^{(k)})$. By Observation 6.2.1, H' is bridgeless. Therefore, by Lemma 6.1.4, $H' \notin \mathcal{V}_k$. Given that H' is a minor of H and that $H \in \mathsf{obs}(\mathcal{V}_k)$, we conclude that $H = H'$. Therefore, $H \in \mathsf{obs}(\mathcal{F}^{(k)})$. ☐

We are actually going to prove in Subsection 6.2.1 that the only bridges that may occur in an obstruction of $\mathcal{V}_k$ are isolated edges. Then, in Subsection 6.2.2, we will prove that any obstruction of $\mathcal{F}^{(k)}$ can be obtained from an obstruction of $\mathcal{V}_k$ by adding edges. See Figure 6.1 for a comparison of the obstructions of $\mathcal{V}_k$ and $\mathcal{F}^{(k)}$ for $k \leq 3$, where the obstructions of $\mathcal{V}_k$ are taken from [52].

6.2.1 Bridges in the obstructions of $\mathcal{V}_k$

In this subsection, we prove the following.

Lemma 6.2.3. *Let* $k \in \mathbb{N}$ *and* $G \in \mathsf{obs}(\mathcal{V}_k)$ *be a graph. Then the connected components of* G *are 2-connected. Therefore, the bridges of* G *are isolated edges.*

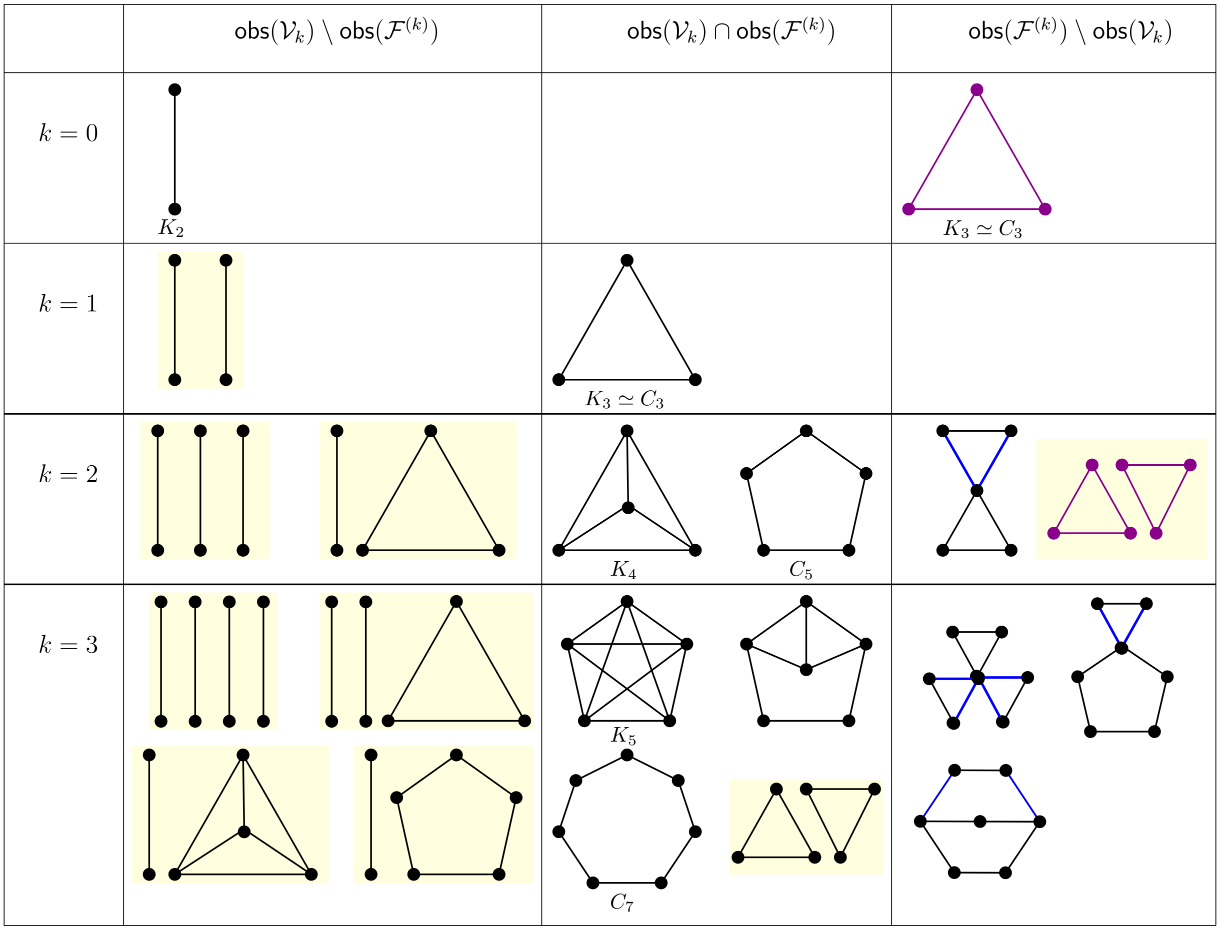

Figure 6.1: The obstructions of $\mathcal{V}_k$ (first and second columns) and $\mathcal{F}^{(k)}$ (second and third columns) for $k \leq 3$. Each graph in $\mathsf{obs}(\mathcal{F}^{(k)})$ is either 1) also a graph in $\mathsf{obs}(\mathcal{V}^{(k)})$ (second column), or 2) can be obtained from a graph in $\mathsf{obs}(\mathcal{V}^{(k)})$ with bridges (first column) by adding edges (in blue in the third column), or 3) is also a graph in $\mathsf{obs}(\mathcal{V}^{(k+1)})$ (in purple in the third column). We use yellow shadows for disconnected obstructions, to make clear that each of them is a single graph.

Actually, we prove a more general version of Lemma 6.2.3 applying on any graph class $\mathcal{H}^{\langle k \rangle}$ defined as follows. Let $\mathcal{H}$ be a hereditary graph class that is also closed under 1-clique-sums. Let $\mathcal{H}^{\langle k \rangle}$ be the set of graphs G such that there exists a set $X \subseteq V(G)$ with $|X| \leq k$ and $G - X \in \mathcal{H}$. In this setting $\mathcal{V}_k = \mathcal{E}^{\langle k \rangle}$, where $\mathcal{E}$ is class of edgeless graphs.

We need the following easy lemma.

Lemma 6.2.4. *Let $\mathcal{H}$ be a hereditary class, $k \in \mathbb{N}$ and $H \in \mathsf{obs}(\mathcal{H}^{\langle k \rangle})$. Then, for any $v \in V(H)$, there is a set $S \subseteq V(H)$ of size $k+1$ such that $v \in S$ and $H - S \in \mathcal{H}$. In particular, $\mathsf{obs}(\mathcal{H}^{\langle k \rangle}) \subseteq \mathcal{H}^{\langle k+1 \rangle} \setminus \mathcal{H}^{\langle k \rangle}$.*

Proof. Let $H \in \mathsf{obs}(\mathcal{H}^{\langle k \rangle})$ and $v \in V(H)$. By definition of an obstruction, $H \notin \mathcal{H}^{\langle k \rangle}$ and $H - \{v\} \in \mathcal{H}^{\langle k \rangle}$. So there is a vertex set S' of size at most k in $H - \{v\}$ such that $H - \{v\} - S' \in \mathcal{H}$. Let $S := S' \cup \{v\}$. Then $H - S \in \mathcal{H}$ so $H \in \mathcal{H}^{\langle k+1 \rangle}$. Given that $H \notin \mathcal{H}^{\langle k \rangle}$, we have $|S| > k$, and therefore, $|S| = k + 1$. $\square$

Here is the main result of the subsection.

Lemma 6.2.5. *Let $k \in \mathbb{N}$. Every connected component of a graph in $\mathsf{obs}(\mathcal{H}^{\langle k \rangle})$ is 2-connected.*

Proof. Suppose towards a contradiction that $G \in \mathsf{obs}(\mathcal{H}^{\langle k \rangle})$ has a connected component that is not 2-connected. Then there is a cut vertex v in G. Let G_1 be a connected component of $G - v$ such that $v \in N_G(V(G_1))$ and let $G_2 = G - V(G_1) - \{v\}$. For $i \in [2]$, let k_i be the minimum k such that $G_i \in \mathcal{H}^{\langle k \rangle}$. Hence, $G_i \in \mathcal{H}^{\langle k_i \rangle} \setminus \mathcal{H}^{\langle k_i - 1 \rangle}$.

Claim 6.2.6. $k = k_1 + k_2$.

Proof. By Lemma 6.2.4, $G \in \mathcal{H}^{\langle k+1 \rangle} \setminus \mathcal{H}^{\langle k \rangle}$. For $i \in [2]$, let $S_i \subseteq V(G_i)$ of size at most k_i be such that $G_i - S_i \in \mathcal{H}$. Then $S := S_1 \cup S_2 \cup \{v\}$ is such that $G - S \in \mathcal{H}$, so $k + 1 \leq k_1 + k_2 + 1$.

By Lemma 6.2.4, there is a set $S \subseteq V(G)$ of size $k+1$ such that $v \in S$ and $G - S \in \mathcal{H}$. Given that $\mathcal{H}$ is hereditary, $G_i - (S \cap V(G_i)) \in \mathcal{H}$. Moreover, $G_i \notin \mathcal{H}^{\langle k_i - 1 \rangle}$ for $i \in [2]$, so we conclude that $|S \cap V(G_i)| \geq k_i$. Hence, $k + 1 = |S| = |\{v\} \cup (S \cap V(G_1)) \cup (S \cap V(G_2))| \geq k_1 + k_2 + 1$. □

For $i \in [2]$, let $\bar{G}_i := G[V(G_i) \cup \{v\}]$. Since $G_i \in \mathcal{H}^{\langle k_i \rangle} \setminus \mathcal{H}^{\langle k_i - 1 \rangle}$ and we only add the vertex v, $\bar{G}_i \in \mathcal{H}^{\langle k_i + 1 \rangle} \setminus \mathcal{H}^{\langle k_i - 1 \rangle}$.

Claim 6.2.7. *There is $i \in [2]$ such that $\bar{G}_i \in \mathcal{H}^{\langle k_i + 1 \rangle} \setminus \mathcal{H}^{\langle k_i \rangle}$.*

Proof. Suppose that $\bar{G}_i \in \mathcal{H}^{\langle k_i \rangle}$ for $i \in [2]$. Let $S_i \subseteq V(\bar{G}_i)$ of size k_i be such that $\bar{G}_i - S_i \in \mathcal{H}$. Then $S := S_1 \cup S_2$ has size at most $k_1 + k_2 < k + 1$. Moreover, given that $\mathcal{H}$ is closed under 1-clique-sums, we have $G - S \in \mathcal{H}$. By Claim 6.2.6, it follows that $G \in \mathcal{H}^{\langle k \rangle}$, a contradiction. □

By Claim 6.2.7, without loss of generality, we assume that $\bar{G}_1 \in \mathcal{H}^{\langle k_1 + 1 \rangle} \setminus \mathcal{H}^{\langle k_1 \rangle}$. Let G' be the graph obtained from the disjoint union of $\bar{G}_1$ and G_2. Given that $\mathcal{H}$ is closed under disjoint union and by Claim 6.2.6, $G' \in \mathcal{H}^{\langle k_1 + 1 + k_2 \rangle} \setminus \mathcal{H}^{\langle k_1 + k_2 \rangle} = \mathcal{H}^{\langle k+1 \rangle} \setminus \mathcal{H}^{\langle k \rangle}$. G' is a subgraph of G so this contradicts the minimality of G as an obstruction of $\mathcal{H}^{\langle k \rangle}$. □

6.2.2 Constructing the obstructions of $\mathcal{F}^{(k)}$ from the obstructions of $\mathcal{V}_k$

What Lemma 6.2.2 and Lemma 6.2.5 tell us is that the difference (as sets) between $\mathsf{obs}(\mathcal{V}_k)$ and $\mathsf{obs}(\mathcal{F}^{(k)})$ is caused by isolated edges. Essentially, to go from an obstruction H of $\mathcal{V}_k$ with isolated edges to an obstruction H' of $\mathcal{F}^{(k)}$, we will have to add vertices and edges minimally to get a bridgeless graph. In this section, we prove that we actually just need to add edges.

Let $\mathsf{Obs} = \bigcup_{k \in \mathbb{N}} \mathsf{obs}(\mathcal{F}^{(k)})$. We have the following easy observation.

Observation 6.2.8. *Let $G \in \mathsf{Obs}$ and $k := \mathsf{idf}(G) - 1$. Then $G \in \mathsf{obs}(\mathcal{F}^{(k)})$.*

Note that, while we observed in Lemma 6.2.4, in particular, that $\mathsf{obs}(\mathcal{V}_k) \subseteq \mathcal{V}_{k+1} \setminus \mathcal{V}_k$, the same does not hold for $\mathcal{F}_k$. For instance, $k \cdot K_3$ (see Figure 6.4) belongs to both $\mathsf{obs}(\mathcal{F}_{2k-2})$ and $\mathsf{obs}(\mathcal{F}_{2k-1})$. However, we can prove the following.

Lemma 6.2.9. *Let $k \in \mathbb{N}$. Then $\mathsf{obs}(\mathcal{F}_k) \subseteq \mathcal{F}_{k+2} \setminus \mathcal{F}_k$.*

Proof. Let $G \in \mathsf{obs}(\mathcal{F}_k)$ and $uv \in E(G)$. Let $\mathcal{X}$ be an id-$\mathcal{F}$ partition of G/uv. Then $G /\!\!/ (u,v) /\!\!/ \mathcal{X} \in \mathcal{F}$, so $G /\!\!/ \mathcal{X}' \in \mathcal{F}$, where $\mathcal{X}'$ is obtained from $\mathcal{X}$ by further identifying u and v. Thus, $|\bigcup \mathcal{X}'| \leq |\bigcup \mathcal{X}| + 2 \leq k + 2$, hence the result. □

The main result of this subsection is the following.

Lemma 6.2.10. *Let G be a graph and $k := \mathsf{idf}(G) - 1$. If $G \in \mathsf{obs}(\mathcal{F}^{(k)})$, then there is $H \in \mathsf{obs}(\mathcal{V}_k)$ that is a minor of G, and for any such H, there is $E' \subseteq E(G)$ such that $G - E' = H$.*

Proof. By Observation 6.2.1, G is bridgeless. Therefore, by Lemma 6.1.4, $\mathsf{idf}(G) = \mathsf{vc}(G)$, and thus $G \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. We first prove that, for any edge $e \in E(G)$, $G/e \in \mathcal{V}_k$.

Claim 6.2.11. *For any edge $uv \in E(G)$, $G/uv \in \mathcal{V}_k$.*

Proof of claim. Suppose towards a contradiction that there is an edge $uv \in E(G)$ such that $G/uv \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. Let w be the heir of uv in G/uv. Since $G \in \mathsf{obs}(\mathcal{F}^{(k)})$, it implies that $G/uv \in \mathcal{F}^{(k)}$. By Observation 6.2.1, G is bridgeless. Thus, by Lemma 6.1.4 and since $G/uv \in \mathcal{F}^{(k)} \setminus V_k$, it implies that the contraction of u and v created a bridge e. Given that only the edges incident to u and v are involved in the contraction, the bridges of G/uv are exactly the edges xw where $x \in N_G(u) \cap N_G(v)$ is a cut vertex of G (the edges xu and xv in G are contracted to xw in G/uv). See Figure 6.2 for an illustration. Let $\mathcal{C}$ be the set of all such x. Let E_1 be the set of all edges xu, xv of G for $x \in \mathcal{C}$ and let E_2 be the set of all edges xw of G/uv for $x \in \mathcal{C}$.

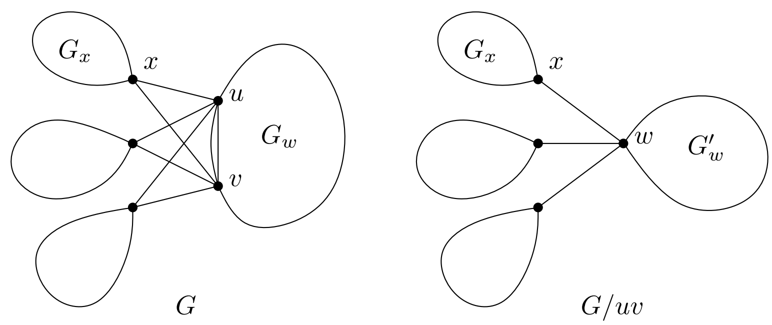

Figure 6.2: Graphs G and G/uv.

Given that $G/uv \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$, there is $H \in \mathsf{obs}(\mathcal{V}_k)$ that is a minor of G/uv. For $x \in \mathcal{C}$, let G_x be the connected component of $G - E_1$ containing x and G_w be the disjoint union of the remaining components of $G - E_1$. Note that G_x is also the connected component of $G/uv - E_2$ containing x for $x \in \mathcal{C}$, and that $G'_w := G_w/uv$ is the union of the other connected components of $G/uv - E_2$. Given that $G/uv - E_2$ is bridgeless, so are G_x for $x \in \mathcal{C}$ and G'_w. By Lemma 6.2.5, each connected component of H is 2-connected. Therefore, given a model M of H of minimal size in G/uv, a bridge of G/uv belongs to M if and only if it is an isolated edge in M. Therefore, H is either a minor of $F := G'_w \cup \bigcup_{x \in \mathcal{C}} G_x$ or, for some $x \in \mathcal{C}$, a minor of $F_x := G[\{x, w\}] \cup (G'_w - \{w\}) \cup (G_x - x) \cup \bigcup_{y \in \mathcal{C} \setminus \{y\}} G_y$. See Figure 6.3 for an illustration.

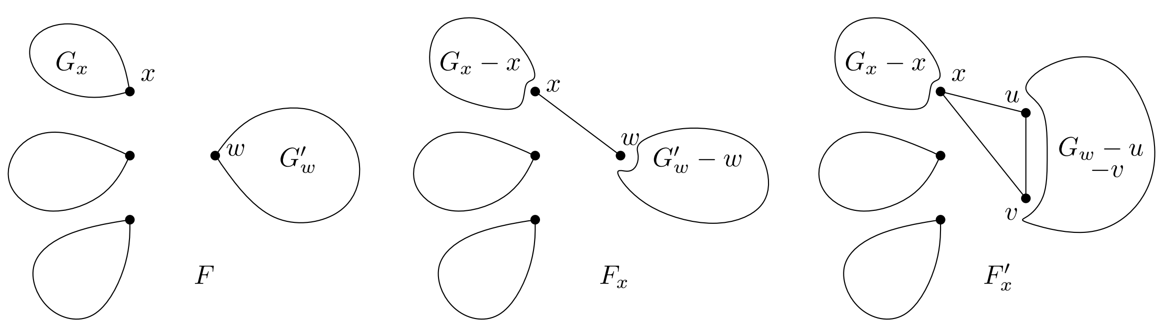

Figure 6.3: Graphs F, F_x, and F'_x.

If H is a minor of F which is a minor of G, then $F \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. Given that F is bridgeless, by Lemma 6.1.4, we thus have $F \in \mathcal{F}^{(k+1)} \setminus \mathcal{F}^{(k)}$. This contradicts the fact that $G \in \mathsf{obs}(\mathcal{F}^{(k)})$.

Hence, H is a minor of F_x for some $x \in \mathcal{C}$. Then H is also a minor of $F'_x := G[\{x,u,v\}] \cup (G_w - u - \{v\}) \cup \cup (G_x - x) \bigcup_{y \in \mathcal{C} \setminus \{x\}} G_y$, which is a minor of G. Thus, $F_x, F'_x \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. Let S be a vertex cover of F'_x of minimum size, i.e., $|S| = k+1$. Let $S' := S \cap \{x,u,v\}$. Given that $G[\{x,u,v\}]$ is a triangle, $|S'| = 2$. But then, $S \setminus S' \cup \{x\}$ is a vertex cover of F_x of size k, a contradiction. ⋄

We now prove that, for any vertex $v \in V(G)$, $G - v \in \mathcal{V}_k$.

Claim 6.2.12. *For any vertex $v \in V(G)$, $G - v \in \mathcal{V}_k$.*

Proof of claim. Suppose towards a contradiction that there is a vertex $v \in \mathcal{V}_k$ such that $G - v \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. If v is an isolated vertex, then $G - v$ is bridgeless. So by Lemma 6.1.4, $G - v \in \mathcal{F}^{(k+1)} \setminus \mathcal{F}^{(k)}$, contradicting that $G \in \mathsf{obs}(\mathcal{F}^{(k)})$. So there is a vertex $u \in N_G(v)$. Let us prove that $G/uv \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. This will contradict Claim 6.2.11 and prove the claim.

Suppose towards a contradiction that $G/uv \in \mathcal{V}_k$. Let S be a vertex cover of G/uv of size k. Let w be the heir of the edge uv in G/uv. If w belongs to S, then $S \setminus \{w\} \cup \{u,v\}$ is a vertex cover of G of size $k+1$ containing v. If w does not belong to S, then $N_{G/uv}(w) \subseteq S$. Since $N_{G/uv}(w) = N_G(\{u,v\})$, we conclude that $S \cup \{v\}$ is a vertex cover of G of size $k+1$ containing v. In both cases, G has a vertex cover S' of size $k+1$ containing v. Therefore, $G - v$ has a vertex cover of size k, contradicting the fact that $G - v \notin \mathcal{V}_k$. ⋄

Given that $G \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$, there is $H \in \mathsf{obs}(\mathcal{V}_k)$ that is a minor of G. By Lemma 6.2.4, $H \in \mathcal{V}_{k+1} \setminus \mathcal{V}_k$. H is obtained from G by a sequence of vertex deletions, edge deletions, and edge contractions such that at each step, the resulting graph belong to $\mathcal{V}_{k+1} \setminus \mathcal{V}_k$. In particular, we can first do all vertex deletions and edge contractions and then the remaining edge deletions. But then, by Claim 6.2.11 and Claim 6.2.12, we cannot do any vertex deletion nor edge contraction and still remain in $\mathcal{V}_{k+1} \setminus \mathcal{V}_k$. Therefore, there is $E' \subseteq E(G)$ such that $G - E' \in \mathsf{obs}(\mathcal{V}_k)$. This concludes the proof. □

We thus have the following upper bound on the size of obstructions, which is a restatement of Theorem 2.2.3.

Theorem 6.2.13. *Let $k \in \mathbb{N}$. For any obstruction $G \in \mathsf{obs}(\mathcal{F}^{(k)})$, $|V(G)| \leq 2k+4$.*

Proof. The obstruction of maximal size in $\mathsf{obs}(\mathcal{V}_k)$ is $(k+1) \cdot K_2$, i.e., the graph obtained from the disjoint union of $k+1$ isolated edges, which has size $2k+2$.

Let $G \in \mathsf{obs}(\mathcal{F}^{(k)})$. By Lemma 6.2.9, we have $\mathsf{idf}(G) \in \{k+1, k+2\}$. Moreover, by Lemma 6.2.10, there is $E' \subseteq E(G)$ such that $G - E' \in \mathsf{obs}(\mathcal{V}_{\mathsf{idf}(G)-1})$. Therefore, $G - E'$, and thus G, has size at most $2k+4$. □

6.3 Universal obstruction

Universal obstruction of idf. Recall that parametric graphs were defined in Section 5.1. We say that two parametric graphs $\mathscr{G}^1$ and $\mathscr{G}^2$ are *comparable* if every graph in $\mathscr{G}^1$ is a minor of a graph in $\mathscr{G}^2$ or every graph in $\mathscr{G}^1$ is a minor of a graph in $\mathscr{G}^2$. Given a minor-monotone graph parameter $\mathsf{p} : \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$, and a finite set $\mathfrak{G} = \{\mathscr{G}^1, \ldots, \mathscr{G}^r\}$ of pairwise non-comparable parametric graphs, we say that $\mathfrak{G}$ is a *universal obstruction* of p if there is a function $f : \mathbb{N} \to \mathbb{N}$ (we refer to f as the *gap function*) such that

- for every $k \in \mathbb{N}$, if G excludes all graphs in $\{\mathscr{G}^1_k, \ldots, \mathscr{G}^r_k\}$ as a minor, then $\mathsf{p}(G) \leq f(k)$.
- $\mathsf{p}(\mathscr{G}^j_k) \geq f(k)$, for every $j \in [r]$.

Universal obstructions serve as asymptotic characterizations of graph parameters, as they identify the typical patterns of graphs that should appear whenever the value of a parameter becomes sufficiently big. Several structural dualities on graph parameters can be described using universal obstructions, and it has been conjectured that for every minor-monotone parameter there always exists some *finite* universal obstruction [247]. (For a survey on universal obstructions see [248].)

Let us give two examples of universal obstructions. A universal obstruction for vc is the set $\{\langle k \cdot K_2 \rangle_{k\in\mathbb{N}}\}$[1] with linear gap function $f(k) = \mathcal{O}(k)$. Another example is the universal obstruction for the parameter fvs, where $\mathsf{fvs}(G)$ is the minimum size of a vertex set of G whose removal yields an acyclic graph. An interpretation of the Erdős-Pósa's theorem [108] is that $\{\langle k \cdot K_3 \rangle_{k\in\mathbb{N}}\}$ is a universal obstruction for fvs with gap function $f(k) = \mathcal{O}(k \cdot \log k)$. Notice that idf can be seen as the analogue of fvs where now, instead of removing vertices, we pick a set of vertices and apply identifications to them.

Our result in this part is a universal obstruction for idf. We use C_k for the cycle on k vertices and $k * K_3$ for the *k-marguerite* graph, that is, the graph obtained from $k \cdot K_3$ by selecting one vertex from each connected component and identifying all selected vertices into a single one (see Figure 6.4).

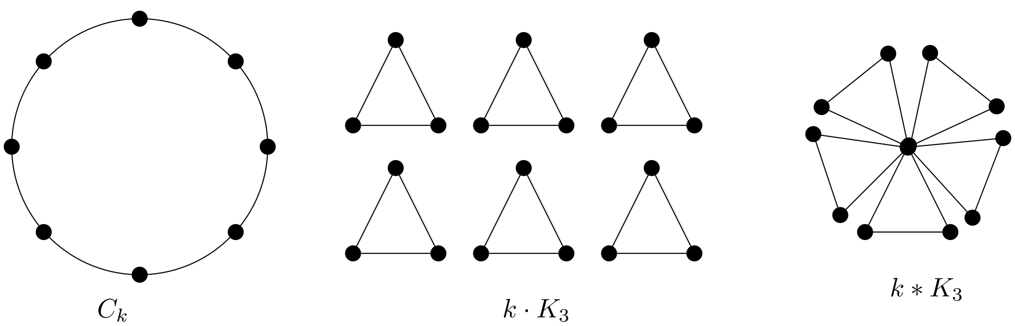

Figure 6.4: The universal obstruction for Identification to Forest.

Theorem 6.3.1. *The set $\{\langle k \cdot K_3 \rangle_{k\in\mathbb{N}}, \langle C_k \rangle_{k\in\mathbb{N}}, \langle k * K_3 \rangle_{k\in\mathbb{N}}\}$ is a universal obstruction of idf, with gap function $f(G) = \mathcal{O}(k^4 \cdot \log^2 k)$.*

One side of the proof is the easy following observation.

Observation 6.3.2. *C_{2k+1}, $\lfloor \frac{k}{2} + 1 \rfloor \cdot K_3$, and $(k+1) * K_3$ are in $\mathsf{obs}(\mathcal{F}^{(k)})$.*

The other side is a bit more involved:

Lemma 6.3.3. *If G excludes every graph in $\{C_k, k \cdot K_3, k * K_3\}$ as a minor, then $\mathsf{idf}(G) = \mathcal{O}(k^4 \cdot \log^2 k)$.*

Proof. Let G be a $\{C_k, k \cdot K_3, k * K_3\}$-minor-free graph. By Lemma 6.1.4, we can assume without loss of generality that G is bridgeless. In particular, any vertex of G has degree at least two.

By the Erdős–Pósa's theorem [108], either G has a packing of k cycles, or there is a set X of size $\mathcal{O}(k \cdot \log k)$ such that $G - X \in \mathcal{F}$. Given that G is $k \cdot K_3$-minor-free, there exists such a set X and $G[X]$ has at most $\mathcal{O}(k \cdot \log k)$ connected components.

Let C be a connected component of $G[X]$. Let $\mathcal{T}_C$ be the set of trees in F with a neighbor in C. Given that G is bridgeless and that any path from a vertex of $T \in \mathcal{T}_C$ to a vertex of $G - V(C) - V(T)$

[1] For a graph H, we denote by $k \cdot H$ the union of k disjoint copies of H.

intersects C, we have $|E_G(V(T), V(C))| \geq 2$. Hence, there is a cycle in the graph induced by T and C. Hence, $|\mathcal{T}_C| * K_3$ is a minor of G. Therefore, $|\mathcal{T}_C| \leq k-1$.

Let $T \in \mathcal{T}_C$. Let T^C be the subtree of T obtained by iteratively removing every leaf of T that is not in $N_G(V(C))$. Hence, for every pair of leaves u, v of T^C, there are two (u,v)-paths P_1 and P_2, the first one in T^C and the second one going through C, that are internally vertex-disjoint. So there is a cycle of length at least $\Delta(T^C)+1$, where $\Delta(T^C)$ denotes the diameter of T^C. Given that C_k is not a minor of G, T^C has diameter at most $k-2$.

Let $L(T^C)$ denote the leaves of T^C, and let $PL(T^C)$ denote the parents of vertices in $L(T^C)$. We claim that $|PL(T^C)| \leq k$. Indeed, let $u \in L(T^C)$ be a leaf picked arbitrarily. Let $V' = V(C) \cup V(T^C) \setminus L(T^C) \setminus PL(T^C) \cup \{u, \mathsf{p}(u)\}$, where $\mathsf{p}(u)$ is the parent of u in T^C. Observe that, since u is connected to C, $G[V']$ is connected. Hence, we can contract V' to a single vertex c to obtain a graph G'. For each $t \in PL(T^C) \setminus \{\mathsf{p}(u)\}$, there is a triangle ctv_t where $v_t \in L(T^C)$ is a child of t. Hence, $(|PL(T^C)|-1) * K_3$ is a subgraph of G' and thus a minor of G. Since $k * K_3$ is not a minor of G, we proved our claim.

Therefore, $|V(T^C) \setminus L(T^C)| \leq \Delta(T^C) \cdot |PL(T^C)| \leq k \cdot (k-2)$.

Let E' be the set of all edges of F that do not belong to T^C for any $C \in \mathsf{cc}(G[X])$ and $T \in \mathcal{T}_C$. Let $e \in E'$. Since e is not a bridge, e is part of a cycle C_e. Hence, there are $C, C' \in \mathsf{cc}(G[X])$ and $T \in \mathcal{T}_C \cap \mathcal{T}_{C'}$ such that any path from T^C to $T_{C'}$ in T goes through e. Moreover, there are at most $k-5$ such edges between T^C to $T_{C'}$, since otherwise C_e would have length at least k. Hence, $|E'| \leq (k-5) \cdot \binom{|\mathsf{cc}(G[X])|}{2} \cdot \max_{C \in \mathsf{cc}(G[X])} |\mathcal{T}_C| = \mathcal{O}(k^4 \cdot \log^2 k)$.

Let $V' \subseteq V(G)$ be the union of X, of the endpoints of edges in E', and of the internal nodes of T^C for any $C \in \mathsf{cc}(G[x])$ and any $T \in \mathcal{T}_C$. Then, $V(G) \setminus V' \subseteq L(F)$, so $G /\!\!/ V'$ is a star. Moreover, $|V| = \mathcal{O}(k \cdot \log k + k^4 \cdot \log^2 k + k \cdot \log k \cdot k \cdot k^2) = \mathcal{O}(k^4 \cdot \log^2 k)$. □

Proof of Theorem 6.3.1. The first condition of the universal obstruction property follows from Lemma 6.3.3 an the second one follows from Observation 6.3.2. □

6.4 Relation with Contraction to $\mathcal{H}$

An important feature of Identification to $\mathcal{H}$ is that it behaves similarly to the problem Deletion to $\mathcal{H}$, in the sense that both problems are FPT when $\mathcal{H}$ is a minor-closed graph class. This follows from Lemma 5.3.1 and the algorithmic consequence of the Robertson and Seymour's theorem [188, 205, 271, 279, 281]. It is easy to observe that the problem Contraction to $\mathcal{H}$ (that is, asking whether k edge contractions yield property $\mathcal{H}$) does not have this property. To see this, let $\mathcal{P}$ be the class of planar graphs and let $K_{3,4}^+$ (resp. $K_{2,3}^+$) be the graph obtained from $K_{3,4}$ (resp. $K_{2,3}$) by adding an edge e between two vertices of degree three (resp. two). Contracting e yields a planar (resp. acyclic) graph, so $(K_{3,4}^+, 1)$ (resp. $(K_{2,3}^+, 1)$) is a yes-instance of Contraction to $\mathcal{P}$ (Contraction to Forest). However, $(K_{3,4}, 1)$ (resp. $(K_{2,3}, 1)$) is a no-instance of the corresponding problem.

Let us define the parameter $\mathsf{ec}_{\mathcal{H}} : \mathcal{G}_{\mathsf{all}} \to \mathbb{N}$, corresponding to the problem Contraction to $\mathcal{H}$, i.e., $\mathsf{ec}_{\mathcal{H}}(G)$ is the minimum number of edge contractions that can transform G to a graph in $\mathcal{H}$. As we observed above, neither $\mathsf{ec}_{\mathcal{F}}$ nor $\mathsf{ec}_{\mathcal{P}}$ are minor-monotone, and similar counterexamples can be found for other instantiations of $\mathcal{H}$. We use ecf as a shortcut for $\mathsf{ec}_{\mathcal{F}}$ and we next observe that idf and ecf are functionally equivalent.

Lemma 6.4.1. *For every graph G it holds that $\mathsf{idf}(G) = \mathcal{O}(\mathsf{ecf}(G))$ and that $\mathsf{ecf}(G) = \mathcal{O}((\mathsf{idf}(G))^3)$.*

Proof. Using the fact that edge contractions are also edge identifications, it easily follows that, for every graph G, $\mathsf{idf}(G) \leq 2 \cdot \mathsf{ecf}(G)$.

Assume now that $\mathsf{idf}(G) \leq k$ and we claim that $\mathsf{ecf}(G) = \mathcal{O}(k^3)$. To prove this claim we first observe that, because $\mathsf{idf}(k \cdot K_3) = \Omega(k)$ and $\mathsf{idf}(k * K_3) = \Omega(k)$ (see Observation 6.3.2), it follows that the number of 2-connected components of G that are not bridges is bounded by some linear function of k. Let B be a 2-connected component of G. As B is a minor of G, it has an id-$\mathcal{F}$ partition $\mathcal{X} = (X_1, \ldots, X_p)$ of order $\leq k$. For $i \in [p]$, let $x^i_1, \ldots, x^i_{p_i}$ be an ordering of the vertices of X_i and let $F_i = \{\{x^i_1, x^i_2\}, \{x^i_2, x^i_3\}, \ldots, \{x^i_{p_i-1}, x^i_{p_i}\}\}$. Let also $F = F_1 \cup \cdots \cup F_p$. Clearly, the 2-element sets in F are not necessarily edges of B. For each $\{x, y\} \in F$ we define a set of edges $F_{x,y}$ as follows. As B is 2-connected, x and y belong to a cycle of B. As $\mathsf{idf}(C_k) = \Omega(k)$ (see Observation 6.3.2), this implies that x and y are joined in B by a path of length $\mathcal{O}(k)$. The edges of this path are the edges in $F_{x,y}$. We now set $F^+ = \bigcup_{\{x,y\} \in F} F_{x,y}$ and observe that $|F^+| = \mathcal{O}(k^2)$. Notice now that contracting the edges of F^+ in B yields an acyclic graph. Therefore, applying these contractions to every non-bridge connected component of G, we obtain an acyclic graph. As there are $\mathcal{O}(k)$ such components, the lemma follows. □

In other words, ecf is not minor-monotone but, however, it is "functionally" monotone in the sense that if G' is a minor of G then $\mathsf{ecf}(G') \leq \mathcal{O}((\mathsf{ecf}(G))^3)$.[2]

6.5 Identification minors

We say that a graph H is an *identification minor* of a graph G if H can be obtained from a minor of G after identifying vertices. As the minor relation between two graphs also implies their identification minor relation, Robertson and Seymour's theorem [278] implies that graphs are well-quasi-ordered by the identification minor relation. It is also easy to observe that, for every graph H, the graphs in the set $\mathcal{M}_H$ of minor-minimal graphs containing H as an identification have size is bounded by a quadratic function of $|H|$. Therefore, checking whether H is an identification minor of G can be done in time $\mathcal{O}_{|H|}(|G|^{1+\varepsilon})$, according to the recent results in [205].

It is a natural question to ask whether graphs are well-quasi-ordered with respect to the vertex identification operation alone. The answer turns out to be negative. Indeed, there is an infinite antichain $(H_k)_{k \in \mathbb{N}}$, where H_k is the graph formed from a cycle on $3k$ vertices $p_1, \ldots, p_{3k}$ by adding three vertices a_1, a_2, a_3 and an edge between each pair (a_i, p_j) such that j is equal to i modulo three. See Figure 6.5 for an illustration. It can be verified that this family of graphs is indeed an antichain, even if we allow both vertex identifications and vertex removals.

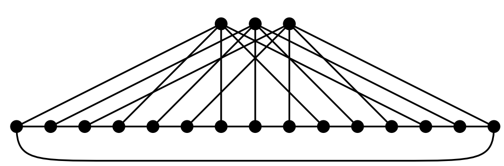

Figure 6.5: The graph H_k for $k = 5$. We give credit to Hugo Jacob for finding it.

We now wish to give the following interpretation of Lemma 6.3.3 in terms of identification minors. To prove it, one needs to observe that $k * K_3$ is an identification-minor of both $k \cdot K_3$ and C_{3k}.

Theorem 6.5.1. *For every graph G and positive integer k, either G contains the k-marguerite $k * K_3$ as an identification minor, or G can become acyclic after applying $\mathcal{O}(k^4 \cdot \log^2 k)$ vertex identifications.*

[2]The cubic bound in Lemma 6.4.1 is just indicative and has not been optimized.

The above theorem can be seen as an analogue of the Erdős-Pósa's theorem [108] where instead of the vertex removal operation we have vertex identification, and instead of $k \cdot K_3$ minor containment we have $k * K_3$ identification minor containment.

CHAPTER 7

Bounded size modifications to minor-closedness

Contents

In this chapter, we prove the results presented in Section 2.3, which are restated here for convenience.

Theorem 2.3.1. *Let $\mathcal{H}$ be a minor-closed graph class and $\mathcal{L}$ be a hereditary replacement action. Then $\mathcal{L}$-*Replacement to *$\mathcal{H}$ can be solved in time $2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}} \cdot n^2$.*

Theorem 2.3.2. *Let $\mathcal{G}_\Sigma$ be the class of graphs embeddable in a surface Σ of Euler genus at most g. Then $\mathcal{L}$-*Replacement to *$\mathcal{G}_\Sigma$ can be solved in time $2^{\mathcal{O}_g(k^9)} \cdot n^2$.*

Theorem 2.3.3. *Let $\mathcal{H}$ be a minor-closed graph class. Then $\mathcal{L}$-*Replacement to *$\mathcal{H}$ can be solved in time $2^{\mathcal{O}(k^2+(k+w)\log(k+w))} \cdot n$ on graphs of treewidth at most w.*

More specifically, Section 7.1 is devoted to the formal definition of the $\mathcal{L}$-Replacement to $\mathcal{H}$ problem ($\mathcal{L}$-R-$\mathcal{H}$) and to examples of problems generated by different instantiations of $\mathcal{L}$. In Section 7.2, we give an overview of the techniques we use and the three main ingredients we need. In Section 7.3, we prove Theorem 2.3.1 and Theorem 2.3.2. The three main ingredients used in Section 7.3 are proved in the last sections. That is, how to find an irrelevant vertex (in Section 7.4), how to find an obligatory set (in Section 7.5), and how to conclude on graphs of bounded treewidth (Theorem 2.3.3, in Section 7.6).

Some conventions and notations. In this chapter and the next, instead of considering a minor-closed graph class $\mathcal{H}$, we consider its obstruction set $\mathcal{F}$, and thus the minor-closed graph class $\mathsf{exc}(\mathcal{F})$. We define three constants depending on $\mathcal{F}$ that are used throughout the chapter whenever we consider such a collection $\mathcal{F}$. We define $a_{\mathcal{F}}$ as the minimum apex number of a graph in $\mathcal{F}$, we set $s_{\mathcal{F}} := \max\{|V(F)| \mid F \in \mathcal{F}\}$, and we define $\ell_{\mathcal{F}}$ to be the maximum detail of a graph in $\mathcal{F}$. Notice that $s_{\mathcal{F}} \leq \ell_{\mathcal{F}} \leq s_{\mathcal{F}}(s_{\mathcal{F}}-1)/2$, and thus $\mathcal{O}_{\ell_{\mathcal{F}}}(\cdot) = \mathcal{O}_{s_{\mathcal{F}}}(\cdot)$.

7.1 Definition of the problem, results, and applications

In this section, we formally define the $\mathcal{L}$-R-$\mathcal{H}$ problem and its annotated version in Subsection 7.1.1. Then, we give in Subsection 7.1.2 a non-exhaustive list of graph modification problems that correspond to different instantiations of $\mathcal{L}$.

7.1.1 Definition of the problem and main results

To handle several modification problems at once, we adapt the vocabulary of Fomin, Golovach, and Thilikos [121], who introduced the notion of replacement action.

Ordered graphs. For the definitions of the next two paragraphs to be correct, we actually need to consider ordered graphs instead of graphs (see the "Graph modifications" paragraph). An *ordered graph* is a graph G equipped with a strict total order on $V(G)$, denoted by $<_G$. In other words, there exists an indexation $v_1, \ldots, v_n$ of the vertices of $V(G)$ such that $v_1 <_G v_2 <_G \cdots <_G v_n$. A subgraph H of an ordered graph G naturally comes equipped with the strict order $<_H$ such that, for each distinct $u, v \in V(H)$, $u <_H v$ if and only if $u <_G v$.

Replacement actions. The *any-replacement action* is the function $\mathcal{M}$ that maps each ordered graph H_1 to the collection $\mathcal{M}(H_1)$ of all the pairs (H_2, ϕ), where H_2 is an ordered graph and $\phi : V(H_1) \to V(H_2) \cup \{\emptyset\}$ is a function such that:

- $|V(H_2)| \leq |V(H_1)|$,
- for each $v \in V(H_2)$, $\phi^{-1}(v) \neq \emptyset$, and
- $<_{H_2}$ is the strict total order such that, for each distinct $v_1, v_2 \in V(H_2)$, we have $v_1 <_{H_2} v_2$ if and only if $u_1 <_{H_1} u_2$ where, for $i \in [2]$, u_i is the smallest vertex (according to $<_G$) in $\phi^{-1}(v_i)$.

A *replacement action* (abbreviated as *R-action*) is any function $\mathcal{L}$ that maps an ordered graph (called a *pattern*) H_1 to a non-empty collection $\mathcal{L}(H_1) \subseteq \mathcal{M}(H_1)$ of its possible *pattern transformations*. See Figure 7.1 for an illustration. The vertices of H_1 mapped by ϕ to the empty set are said to be *deleted*, and two vertices of H_1 mapped by ϕ to the same vertex of H_2 are said to be *identified*. Given $S \subseteq V(H_1)$, we set $\phi^+(S) = \phi(S) \setminus \{\emptyset\}$. Note that, if $\phi(S) = \{\emptyset\}$, then $\phi^+(S) = \{\emptyset\} \setminus \{\emptyset\} = \emptyset$.

Graph modifications. Let $\mathcal{L}$ be an R-action, let G be an ordered graph, and $S \subseteq V(G)$. Let $(H_2, \phi) \in \mathcal{L}(G[S])$. We denote by $G^S_{(H_2,\phi)}$ the graph obtained from the disjoint union of $G - S$ and H_2 by adding an edge $u\phi(v)$ for each $u \in V(G) \setminus S$ and each $v \in \phi^{-1}(V(H_2))$ such that $uv \in E(G)$. We equip $G' := G^S_{(H_2,\phi)}$ with the strict total order $<_{G'}$ such that $v_1 <_{G'} v_2$ if and only if $u_1 <_G u_2$ where, for $i \in [2]$, $u_i := v_i$ if $v_i \in V(G) \setminus S$, and u_i is the smallest vertex in $\phi^{-1}(v_i)$ if $v_i \in V(H_2)$. We also set $\mathcal{L}_S(G) = \{G^S_{(H_2,\phi)} \mid (H_2, \phi) \in \mathcal{L}(G[S])\}$. See Figure 7.1 for an illustration.

Note that we consider ordered graphs merely so that the correspondence between the vertices in S and the vertices in $V(H_2)$ is well-defined. We actually omit the order from the statements, but it will be implicitly assumed that vertices have a label that allows us to keep track of them during the modification procedure.

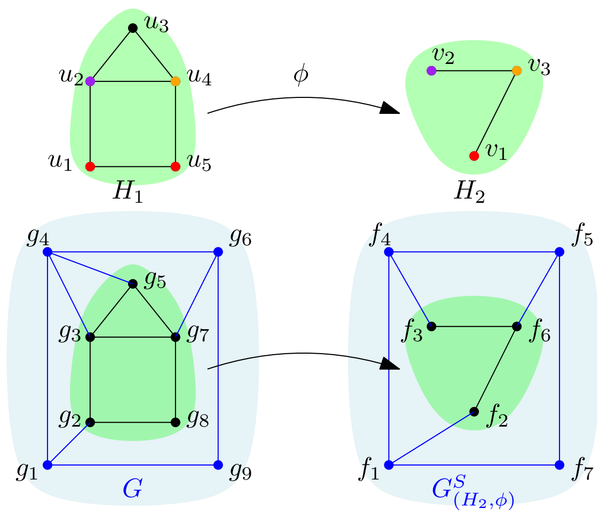

Figure 7.1: Example of an element (H_2, ϕ) in the collection $\mathcal{L}(H_1)$ and of the modified graph $G^S_{(H,\phi)}$ where S is the set of black vertices of G. ϕ is represented by the colors, that is, $\phi(u_1) = \phi(u_5) = v_1$, $\phi(u_2) = \phi(v_2)$, $\phi(u_3) = \emptyset$, and $\phi(u_4) = v_3$. The order on the vertex sets of the depicted graphs is given by the corresponding labels.

Let $\mathcal{L}$ be an R-action and $\mathcal{H}$ be a graph class. We define the problem $\mathcal{L}$-Replacement to $\mathcal{H}$ as follows.

$\mathcal{L}$-Replacement to $\mathcal{H}$ ($\mathcal{L}$-R-$\mathcal{H}$)

Input: A graph G and $k \in \mathbb{N}$.

Question: Is there a set $S \subseteq V(G)$ of size at most k such that $\mathcal{L}_S(G) \cap \mathcal{H} \neq \emptyset$?

Such a set S is called *solution* of $\mathcal{L}$-R-$\mathcal{H}$ for the instance (G, k).

Let us observe the following, which implies that a no-instance for Vertex Deletion to $\mathcal{H}$ is also a no-instance for $\mathcal{L}$-R-$\mathcal{H}$.

Observation 7.1.1. *Let $\mathcal{H}$ be a hereditary graph class, let $\mathcal{L}$ be an R-action, let G be a graph, and let $S \subseteq V(G)$. If $\mathcal{L}_S(G) \cap \mathcal{H} \neq \emptyset$, then $G - S \in \mathcal{H}$.*

Proof. Indeed, suppose that there is $(H_2, \phi) \in \mathcal{L}(G[S])$ such that $G^S_{(H_2,\phi)} \in \mathcal{H}$. Then, because $\mathcal{H}$ is hereditary, $G^S_{(H_2,\phi)} - \phi^+(S) = G - S \in \mathcal{H}$. □

To find a wall quickly in a graph, we can hence use the following proposition.

Proposition 7.1.2 ([284]). *Let $\mathcal{F}$ be a finite collection of graphs. There exist a function $f_{7.1.2} : \mathbb{N} \to \mathbb{N}$ and an algorithm with the following specifications:*

`Find-Wall`(G, r, k)
Input: *A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, and $k \in \mathbb{N}$.*
Output: *One of the following:*

- **Case 1:** *Either a report that (G, k) is a* no*-instance of* Vertex Deletion to exc$(\mathcal{F})$, *or*
- **Case 2:** *a report that G has treewidth at most $f_{7.1.2}(s_{\mathcal{F}}) \cdot r + k$, or*
- **Case 3:** *an r-wall W of G.*

Moreover, $f_{7.1.2}(s_{\mathcal{F}}) = 2^{\mathcal{O}(s_{\mathcal{F}}^2 \cdot \log s_{\mathcal{F}})}$, and the algorithm runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r^2+(k+r)\cdot\log(k+r))} \cdot n$.

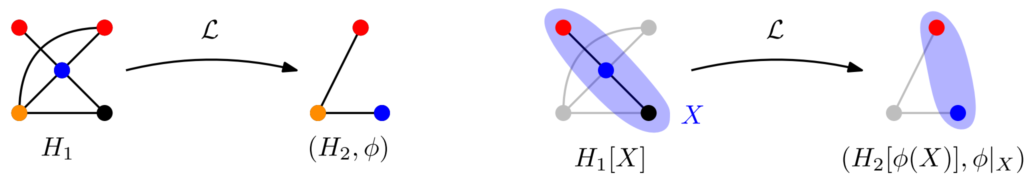

Figure 7.2: If $\mathcal{L}$ is hereditary, then a restriction of an allowed modification is also allowed.

Hereditary R-actions. An R-action is said to be *hereditary* if, for each ordered graph H_1, for each non-empty $X \subseteq V(H_1)$, and for each $(H_2, \phi) \in \mathcal{L}(H_1)$, we have $(H_2[\phi^+(X)], \phi|_X) \in \mathcal{L}(H_1[X])$. We say that $(H_2[\phi^+(X)], \phi|_X)$ is the *restriction* of (H_2, ϕ) to X. See Figure 7.2 for an illustration.

Informally, an R-action is hereditary if, when a modification is allowed, then modifying "less" is allowed as well. For instance, if $\mathcal{L}$ allows us to delete *exactly* k vertices, then $\mathcal{L}$ also allows us to delete *at most* k vertices.

Our main result is the following, which is a restatement of Theorem 2.3.1.

Theorem 7.1.3. *Let $\mathcal{F}$ be a finite collection of graphs and let $\mathcal{L}$ be a hereditary R-action. There is an algorithm that, given a graph G and $k \in \mathbb{N}$, runs in time $2^{\mathsf{poly}_{\mathcal{F}}(k)} \cdot n^2$ and either outputs a solution of $\mathcal{L}$-R-exc$(\mathcal{F})$ for the instance (G, k) or reports a* no*-instance. Moreover,* $\mathsf{poly}_{\mathcal{F}}$ *is a polynomial whose degree depends on the maximum detail of a graph in $\mathcal{F}$.*

As mentioned in the introduction, the main result in [287] already implies that $\mathcal{L}$-R-$\mathcal{H}$ is solvable in time $f(k) \cdot n^2$ when $\mathcal{H}$ is minor-closed for some huge function f that is not even estimated. Our main contribution is an explicit and single-exponential dependence on k (restatement of Theorem 2.3.2).

The degree of $\mathsf{poly}_{\mathcal{F}}(k)$ is quite big, but we can reduce it in some specific cases.

Theorem 7.1.4. *Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{H}$ be the class of graphs embeddable in a surface Σ of Euler genus at most g. There is an algorithm that, given a graph G and $k \in \mathbb{N}$, runs in time $2^{\mathcal{O}_g(k^9)} \cdot n^2$ and either outputs a solution of $\mathcal{L}$-R-$\mathcal{H}$ for the instance (G, k) or reports a* no*-instance.*

More generally, we study the annotated version of $\mathcal{L}$-R-$\mathcal{H}$. Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{H}$ be a graph class. We define the problem $\mathcal{L}$-Annotated Replacement to $\mathcal{H}$ as follows.

$\mathcal{L}$-Annotated Replacement to $\mathcal{H}$ ($\mathcal{L}$-AR-$\mathcal{H}$)

Input: A graph G, a set of annotated vertices $S' \subseteq V(G)$, $(H_2', \phi') \in \mathcal{L}(G[S'])$, and $k \in \mathbb{N}$.

Question: Is there a set $S \subseteq V(G)$ of size at most k and $(H_2, \phi) \in \mathcal{L}(G[S])$ such that (H_2', ϕ') is the restriction of (H_2, ϕ) to S' and $G^S_{(H_2,\phi)} \in \mathcal{H}$?

Obviously, we must have $S' \subseteq S$. Such a triple (S, H_2, ϕ) is called a *solution* of $\mathcal{L}$-AR-$\mathcal{H}$ for the instance (G, S', H_2', ϕ', k). An instance of $\mathcal{L}$-AR-$\mathcal{H}$ where $S' = \emptyset$ is an instance of $\mathcal{L}$-R-$\mathcal{H}$, so $\mathcal{L}$-AR-$\mathcal{H}$ generalizes $\mathcal{L}$-R-$\mathcal{H}$. Two instances $\mathcal{I}_1$ and $\mathcal{I}_2$ are said to be *equivalent* instances of $\mathcal{L}$-AR-$\mathcal{H}$ if $\mathcal{I}_1$ is a yes-instance of $\mathcal{L}$-AR-$\mathcal{H}$ if and only if $\mathcal{I}_2$ is a yes-instance of $\mathcal{L}$-AR-$\mathcal{H}$.

In fact, the results that we actually prove are the following.

Theorem 7.1.5. *Let $\mathcal{F}$ be a finite collection of graphs and let $\mathcal{L}$ be a hereditary R-action. There is an algorithm that, given a graph G, $S' \subseteq V(G)$, $(H_2', \phi') \in \mathcal{L}(G[S'])$, and $k \in \mathbb{N}$, runs in time $2^{\mathsf{poly}_{\mathcal{F}}(k)} \cdot n^2$ and either outputs a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k) or reports a* no*-instance. Moreover, $\mathsf{poly}_{\mathcal{F}}$ is a polynomial whose degree depends on the maximum detail of a graph in $\mathcal{F}$.*

Theorem 7.1.6. *Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{H}$ be the class of graphs embeddable in a surface Σ of Euler genus at most g. There is an algorithm that, given a graph G, $S' \subseteq V(G)$, $(H_2', \phi') \in \mathcal{L}(G[S'])$, and $k \in \mathbb{N}$, runs in time $2^{\mathcal{O}_g(k^9)} \cdot n^2$ and either outputs a solution of $\mathcal{L}$-AR-$\mathcal{H}$ for the instance (G, S', H_2', ϕ', k) or reports a* no*-instance.*

7.1.2 Problems generated by different instantiations of $\mathcal{L}$

Many graph modification problems correspond to $\mathcal{L}$-R-$\mathcal{H}$ for a specific R-action $\mathcal{L}$ and a specific target graph class $\mathcal{H}$. We give a few examples below. Let $\mathcal{H}$ be a minor-closed graph class. For instance, $\mathcal{H}$ could be the class of edgeless graphs, of forests, of graphs whose connected components have size at most k, of planar graphs, or of graphs embeddable in a surface Σ. Note that we do not mention Edge Addition to $\mathcal{H}$ (nor Edge Edition to $\mathcal{H}$) here, because when $\mathcal{H}$ is a minor-closed graph class, adding edges is "unnecessary", in the sense that the edge deletion variant has the same expressive power, and we can solve it. Note also that $\mathcal{L}$-R-$\mathcal{H}$, and thus in particular all problems of this section, was already known to be solvable in FPT-time (when $\mathcal{H}$ is minor-closed) by the result of [287]. However, as mentioned before, the parametric dependence is huge and not even explicit in [287].

Vertex Deletion to $\mathcal{H}$

Input: A graph G and $k \in \mathbb{N}$.

Question: Is there a set $S \subseteq V(G)$ of size at most k such that $G - S \in \mathcal{H}$?

Vertex Deletion to $\mathcal{H}$ reduces to $\mathcal{L}_{\mathsf{vDel}}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{vDel}}$ is the function that maps any graph H_1 to the singleton containing the empty graph and the constant function $\phi : V(H_1) \to \{\emptyset\}$.

Vertex Deletion to $\mathcal{H}$ is already known [235] to be solvable within the same running time as the one of Theorem 7.1.3. Hence, the result of Theorem 7.1.3 is not an improvement for this specific problem, but it shows that our result is tight compared to the currently best known result for Vertex Deletion to $\mathcal{H}$.

Edge Deletion to $\mathcal{H}$

Input: A graph G and $k \in \mathbb{N}$.
Question: Is there a set $F \subseteq E(G)$ of size at most k such that $G - F \in \mathcal{H}$?

(G, k) is a yes-instance of Edge Deletion to $\mathcal{H}$ if and only if $(G, 2k)$ is a yes-instance of $\mathcal{L}_{\mathsf{eDel},k}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{eDel},k}$ is the function that maps each graph H_1 to the set of pairs $(H_1 - F, \mathsf{id}_{V(H_1)})$ over all $F \subseteq E(G)$ of size at most k.

Algorithms with a nice parametric dependence are only known for specific target classes $\mathcal{H}$. Namely, when $\mathcal{H}$ is the class of forests, Edge Deletion to $\mathcal{H}$ corresponds to Feedback Edge Set, which can be solved in polynomial time as mentioned in Section 1.2. When $\mathcal{H}$ is the class of graphs that are a union of paths, then there is a linear kernel for the problem [219], as well as a FPT algorithm with parametric dependence on k at most 2^k [307]. We refer the reader to the survey of [69], as well as [99], for other results with explicit dependence on k when $\mathcal{H}$ is not a minor-closed graph class.

Given a graph G and a set of edges $F \subseteq E(G)$, we denote by G/F the graph obtained from G after contracting the edges in F.

Edge Contraction to $\mathcal{H}$

Input: A graph G and $k \in \mathbb{N}$.
Question: Is there a set $F \subseteq E(G)$ of size at most k such that $G/F \in \mathcal{H}$?

(G, k) is a yes-instance of Edge Contraction to $\mathcal{H}$ if and only if $(G, 2k)$ is a yes-instance of $\mathcal{L}_{\mathsf{Con},k}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{Con},k}$ is the function that maps each graph H_1 to the set of pairs $(H_1/F, \phi)$ over all $F \subseteq E(G)$ of size at most k, where ϕ maps $v \in V(H_1)$ to the corresponding vertex of H_1/F.

An explicit parametric dependence was given in [165] when $\mathcal{H}$ is a class of paths (running time $2^{k+o(k)} + n^{\mathcal{O}(1)}$) or the class of trees (running time $4.98^k \cdot n^{\mathcal{O}(1)}$). Though these classes are not minor-closed, we can easily extend these results to the case when $\mathcal{H}$ is the class of unions of paths or the class of forests (up to a 2^k factor). FPT-algorithms with an explicit parametric dependence were also studied when $\mathcal{H}$ is a collection of generalization and restriction of trees [8, 11], or when $\mathcal{H}$ is the class of cactus graphs [210]. We refer the reader to survey in [147] for more results when the target class is not minor-closed.

Identification to $\mathcal{H}$

Input: A graph G and $k \in \mathbb{N}$.
Question: Is there a set $S \subseteq V(G)$ of size at most k and a partition $(X_1, \dots, X_p)$ of S such that the graph obtained after identifying the vertices in X_i to a single vertex x_i, for $i \in [p]$, belongs to $\mathcal{H}$?

Identification to $\mathcal{H}$ reduces to $\mathcal{L}_{\mathsf{Id}}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{Id}}$ is the function that maps each graph H_1 to the set of pairs (H_2, ϕ), where H_2 can be obtained from H_1 after identifying each X_i of a partition $(X_1, \dots, X_p)$ of some set $S \subseteq V(H_1)$ to a single vertex x_i, and ϕ maps vertices of X_i to x_i and is the identity on $V(H_1) \setminus S$.

We provide in Chapter 6 an FPT-algorithm and a kernel of size $2k + 1$ for Identification to $\mathcal{H}$ when $\mathcal{H}$ is the class of forests. To our knowledge, this is the only known result for this problem.

Independent Set Deletion to $\mathcal{H}$

Input:	A graph G and $k \in \mathbb{N}$.
Question:	Is there an independent set $I \subseteq V(G)$ of size at most k such that $G - I \in \mathcal{H}$?

Independent Set Deletion to $\mathcal{H}$ reduces to $\mathcal{L}_{\mathsf{ISDel}}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{ISDel}}$ is the function that maps any graph H_1 to the set of pairs $(H_1 - I, \phi)$ over all independent sets $I \subseteq V(H_1)$, where ϕ maps vertices of I to the empty set and is the identity on $V(H_1) \setminus I$.

When $\mathcal{H}$ is the class of forests, the problem is known to be solvable in time $3.62^k \cdot n^{\mathcal{O}(1)}$ [218]. Concerning other target classes that are not minor-closed, mainly bipartite graphs, let us mention [5, 39, 134].

To illustrate the versatility of $\mathcal{L}$-R-$\mathcal{H}$, let us present some other problems that can be defined by particular hereditary R-actions, though they do not seem to have been studied when parameterized by the solution size.

(Induced) Matching Deletion to $\mathcal{H}$

Input:	A graph G and $k \in \mathbb{N}$.
Question:	Is there an (induced) matching $M \subseteq E(G)$ of size at most k such that $G - M \in \mathcal{H}$?

(G, k) is a yes-instance of (Induced) Matching Deletion to $\mathcal{H}$ if and only if $(G, 2k)$ is a yes-instance of $\mathcal{L}_{\mathsf{mDel},k}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{mDel},k}$ is defined similarly to $\mathcal{L}_{\mathsf{eDel},k}$ above, but for (induced) matchings.

There are some results on Matching Deletion to $\mathcal{H}$ when $k = n$ and $\mathcal{H}$ is the class of forests [222, 251] or bipartite graphs (see [221] for a small survey on the subject).

(Induced) Matching Contraction to $\mathcal{H}$

Input:	A graph G and $k \in \mathbb{N}$.
Question:	Is there an (induced) matching $M \subseteq E(G)$ of size at most k such that $G/M \in \mathcal{H}$?

(G, k) is a yes-instance of (Induced) Matching Contraction to $\mathcal{H}$ if and only if $(G, 2k)$ is a yes-instance of $\mathcal{L}_{\mathsf{mCon},k}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{mCon},k}$ is defined similarly to $\mathcal{L}_{\mathsf{Con},k}$ above, but for (induced) matchings.

Induced Star Deletion to $\mathcal{H}$

Input:	A graph G and $k \in \mathbb{N}$.
Question:	Is there a set $F \subseteq E(G)$ inducing a star $K_{1,k'}$ with $k' \leq k$ such that $G - F \in \mathcal{H}$?

(G, k) is a yes-instance of Star Deletion to $\mathcal{H}$ if and only if $(G, k+1)$ is a yes-instance of $\mathcal{L}_{\mathsf{StarDel},k}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{StarDel},k}$ is the function that maps any graph H_1 to the set of pairs $(H_1 - F, \mathsf{id}_{V(H_1)})$ over all sets $F \subseteq E(G)$ inducing a subgraph of $K_{1,k}$.

Given a graph G, the *complement* of G, denoted by $\overline{G}$, is graph with vertex set $V(G)$ and edge set the edges that do not belong to $E(G)$.

> **Subgraph Complementation to $\mathcal{H}$**
>
> *Input:* A graph G and $k \in \mathbb{N}$.
>
> *Question:* Is there a set $S \subseteq V(G)$ of size at most k such that the graph obtained after replacing $G[S]$ with its complement $\overline{G[S]}$ belongs to $\mathcal{H}$?

Subgraph Complementation to $\mathcal{H}$ reduces to $\mathcal{L}_{\mathsf{Comp}}$-R-$\mathcal{H}$, where $\mathcal{L}_{\mathsf{Comp}}$ is the function that maps any graph H_1 to the singleton containing the pair $(\overline{H_1}, \mathsf{id}_{V(H_1)})$.

The problem got recently studied when $k = n$ for various target classes. We refer the reader to [14] for a survey on the subject.

Remark. Note that some of the R-actions $\mathcal{L}$ corresponding to a graph modification problem above depend on the parameter k. This implies that the corresponding algorithm is *non-uniform* in k. However, this is just an illusion due to the way we define $\mathcal{L}$-R-$\mathcal{H}$ so that it generalizes all problems at once: we quantify on the size of the set S of modified vertices, while some problems may use a different quantification, such as the number of modified edges. Given a specific graph modification problem Π to $\mathcal{H}$, we can easily tune the algorithms of this chapter so that they work exactly for the modification and the quantification we consider, and in this case, the algorithm would be *uniform* in k.

7.2 Overview of the techniques

We now proceed to provide a high-level overview of the main tools used to prove our results, without getting into technical details. This chapter generalizes the techniques recently introduced in [284] in order to deal with Vertex Deletion to $\mathcal{H}$, which are based on exploiting the Flat Wall Theorem of Robertson and Seymour [271], namely the version proved by Kawarabayashi, Thomas, and Wollan [194] and its recent restatement by Sau, Stamoulis, and Thilikos [286]. Recall from Section 3.1 that the idea of Theorem 2.3.1 (cf. Theorem 7.1.5) and Theorem 2.3.2 (cf. Theorem 7.1.6) is that, as far as the treewidth of the input graph is sufficiently large as an appropriate function of k, it is possible to either "branch" into a number of subproblems that depends only on k and where the value of the parameter is strictly smaller, or to find an irrelevant vertex (i.e., a vertex that does not change the answer to the considered problem) and remove it from the graph. Once the treewidth is bounded, what remains is to apply the most efficient possible algorithm to solve the problem via dynamic programming on tree decompositions.

Let us focus more particularly on the techniques we use to prove Theorem 2.3.1. Contrary to the algorithm of [284] that solves Vertex Deletion to $\mathcal{H}$ for any minor-closed class $\mathcal{H}$, we *avoid using iterative compression*. This explains the improvement from cubic to quadratic complexity in n. The algorithm of Theorem 2.3.1 can be seen as an extension of the algorithm of [284] that solves Vertex Deletion to $\mathcal{H}$ in the particular case where $\mathcal{H}$ is apex-minor-free.

In a nutshell, our algorithm employs a win/win strategy that proceeds as follows:

- If the treewidth of the input graph is small (as a function of the parameter k), then solve the problem via a dynamic programming approach.
- If the treewidth of the input graph is big, then either
  - (*irrelevant vertex*) find a vertex v such that (G, k) and $(G - v, k)$ are equivalent instances, or

- ($branching$ $case$) find a set $A \subseteq V(G)$ of small size such that there exists $v \in A$ such that (G,k) and $(G-v,k-1)$ are equivalent instances,

and recurse.

Hence, we require three ingredients: one to solve the problem parameterized by treewidth, one to find an irrelevant vertex, and one to find an "obligatory set" A, all with a "reasonable" parametric dependence on k. Then, we need to construct an algorithm so that one of these three cases always applies and such that the overall running time is still within the desired bound, which is one of the most convoluted parts of the proof.

Let S' be the set of vertices recursively guessed to be modified in the branching step. An advantage when the modification consists in vertex deletion is that we can simply recurse on $(G-S', k-|S'|)$. For the more general case of $\mathcal{L}$-R-$\mathcal{H}$, we cannot simply delete S', as the considered modification may be different from vertex deletion. We need 1) to guess how $G[S']$ is modified, that is, to guess $(H_2', \phi') \in \mathcal{L}(G[S'])$ and 2) to remember S' and (H_2', ϕ') in order to check that we eventually find a set $S \supseteq S'$ and an allowed modification $(H_2, \phi) \in \mathcal{L}(G[S])$ whose restriction to S' is (H_2', ϕ') such that the modified graph is in $\mathcal{H}$. This is why we need to solve the *annotated version of the problem*, denoted by $\mathcal{L}$-AR-$\mathcal{H}$, where we add to the input a subset S' of vertices of G that are required to be part of H_1, as well as the modification (H_2', ϕ') made on S'.

As for solving the problem when the graph has bounded treewidth, we cannot just use Courcelle's theorem [67], since we require a nice parametric dependence on k. Hence, we need to design our own dynamic programming algorithm to solve $\mathcal{L}$-AR-$\mathcal{H}$ parameterized by the treewidth and k (Theorem 7.3.4, proved in Section 7.6). Essentially, the idea is to guess, in each bag $\beta(t)$ of the decomposition, the set S_t of vertices that are modified as well as how they are modified, and to reduce the size of the graph G_t induced by the bag t and its children using the representative-based technique of [24]. Recall from Section 4.4 that this technique is based on the property that (cf. Proposition 4.4.1 and Proposition 4.4.2), given a boundaried graph $\mathbf{G}$ whose underlying graph belong to a minor-closed graph class $\mathcal{H}$, there is a boundaried graph $\mathbf{R}$ of *bounded size* compatible with $\mathbf{G}$, called the *representative of* $\mathbf{G}$, such that, for any boundaried graph $\mathbf{H}$ compatible with $\mathbf{G}$, we have $\mathbf{G} \oplus \mathbf{H} \in \mathcal{H}$ if and only if $\mathbf{R} \oplus \mathbf{H} \in \mathcal{H}$. G_t does not belong to $\mathcal{H}$, so we cannot find a representative of G_t (with boundary $\beta(t)$), but we find instead a representative of the graph $G_t' \in \mathcal{H}$ modified from G_t according to the guessed modification on S_t and the previously guessed modification on the children of t. Given that we may need to identify together vertices that are far apart in the tree decomposition, we need to remember throughout the algorithm the vertices that are guessed to be part of the solution. The fact that we keep information about these at most k vertices explains the dependence on k of the dynamic programming algorithm, which runs in time $2^{\mathcal{O}(k^2+(k+\mathsf{tw})\log(k+\mathsf{tw}))} \cdot n$.

As expected, finding an irrelevant vertex (Theorem 7.3.1, proved in Section 7.4) is done using the irrelevant vertex technique of Robertson and Seymour [271]. More specifically, we generalize the irrelevant vertex technique used in [285] (cf. Proposition 8.2.4). While our irrelevant vertex technique for $\mathcal{L}$-AR-$\mathcal{H}$ takes inspiration from [285], it is far more involved due to the annotation and the fact that we allow a wide variety of modifications. The fact that we ask the replacement action $\mathcal{L}$ to be hereditary comes from the irrelevant vertex technique. Indeed, in order to prove that the central vertex v of a homogeneous flat wall W is irrelevant, we essentially prove that, for any solution (S, H_2, ϕ), we can delete a small part X of W containing v, and that the restriction of (S, H_2, ϕ) to $G - X$ is still a solution.

The branching case (Lemma 7.3.3, proved in Section 7.5) is not much different from what is done in [285] (cf. Proposition 8.2.5, see also [229, 284]): essentially, if there is a big enough wall W (cf. Figure 3.1) and a set A of vertices having many disjoint paths to W (cf. Figure 7.3), then some modification (H_A, ϕ_A) must happen in A and we can branch. Here, we however need to additionally prove that we must have $|\phi_A(A) \setminus \{\emptyset\}| < |A|$. We stress that it is important here to guess some modification in A that strictly decreases the size of A, so that, after applying this partial modification to G at the next step in the recursion, we will not find the exact same obligatory set A. Hence, in the algorithm with input (G, S', H'_2, ϕ', k), at each step, either we find an irrelevant vertex and strictly decrease the size of G, or we branch and strictly increase the size of S'.

Finally, in Section 7.3 we combine these three ingredients to find an algorithm for $\mathcal{L}$-AR-$\mathcal{H}$. It essentially proceeds as follows. Let (G, S', H'_2, ϕ', k) be the instance we want to solve, and G' be obtained by doing the modification (H'_2, ϕ') of S'. In the first steps, we either find that G has small treewidth, where we can use our dynamic programming algorithm to conclude, or that G' contains a wall W. Given W, we first try to find a flat wall W' inside, with all the necessary conditions to find an irrelevant vertex. If we manage to do so, we remove the irrelevant vertex and recurse. Otherwise, through a greedy procedure, we try to find an obligatory vertex set A with many disjoint paths to W in G'. If we find such a set, we branch and recurse. If not, we manage to argue that we must have a no-instance, and conclude.

The second algorithm (Theorem 2.3.2), when $\mathcal{H}$ is a class of graphs embeddable in a surface of bounded Euler genus, uses two additional ideas to get an improved running time. The first one is that here, the obligatory set A is a singleton. Indeed, the size of A is the size of the minimum number of vertices one can remove from an obstruction of $\mathcal{H}$ to make it planar. It is well known that, when $\mathcal{H}$ is such a class, there is some integer t depending on the Euler genus such that $K_{3,t} \notin \mathcal{H}$ [233, Theorem 4.4.7], and thus, $|A| = 1$. In particular, this implies that we do not need to branch on A, but that we instead immediately find an obligatory vertex. The second idea is about homogeneous flat walls. In the running time $2^{\mathsf{poly}(k)} \cdot n^2$ of the first algorithm, the degree of poly essentially corresponds to the size of the required flat wall to find a big enough homogeneous flat wall, and hence an irrelevant vertex, inside of it. In the case where $\mathcal{H}$ is the class of graphs embeddable in a surface of Euler genus at most g, we prove that we can find a homogeneous flat wall inside a flat wall of smaller size, hence the improved running time (Theorem 7.3.2, proved in Subsection 7.4.3). To do so, we prove that, after some preliminary processing, a flat wall that is furthermore embeddable in a disk with the perimeter on its boundary is already homogeneous (Lemma 7.4.1). Hence, our second algorithm (Subsection 7.3.3) proceeds similarly to the first one, but if we find a flat wall W' in G', we divide W' into $k+1$ disjoint smaller flat walls and check whether they belong to $\mathcal{H}$. By the pigeonhole principle, one of them, W_i, does not contain a modified vertex and must thus be in $\mathcal{H}$, otherwise we return a no-instance. Then, we argue, using a result from [86] (Proposition 7.3.9) to guarantee additional properties of the planar embedding that are needed for technical reasons, that we can find a smaller flat wall W'_i in W_i with a *planar embedding* (even if the genus of the target graph class is strictly positive). Hence, we find an irrelevant vertex in W'_i and conclude.

7.3 The algorithms

In this section, we provide our two algorithms (Theorem 2.3.1 and Theorem 2.3.2). In Subsection 7.3.1, we state the three main ingredients necessary for the algorithms, that will be provided in later sections. In Subsection 7.3.2, we give the algorithm for the general case. In Subsection 7.3.3, we explain how to improve the algorithm in the special case where $\mathcal{H}$ is the class of graphs embeddable in a surface of bounded genus.

7.3.1 Main ingredients

The first ingredient is a result stating that an irrelevant vertex can be found in a big enough flat wall whose compass has bounded treewidth. The proof is deferred to Subsection 7.4.2.

Theorem 7.3.1. *Let $\mathcal{F}$ be a finite collection of graphs and $\mathcal{L}$ be a hereditary R-action. There exist a function $f_{7.3.1} : \mathbb{N}^2 \to \mathbb{N}$, whose images are odd integers, and an algorithm with the following specifications:*

`Irrelevant-Vertex`$(G, S', H'_2, \phi', k, A, a, W, \mathfrak{R}, t)$
Input: *Integers $k, a, t \in \mathbb{N}$, a graph G, a set $S' \subseteq V(G)$ of size at most k, $(H'_2, \phi') \in \mathcal{L}(G[S'])$, a set $A \subseteq V(G')$ of size at most a, where $G' := G^{S'}_{(H_2,\phi)}$, and a regular flatness pair $(W, \mathfrak{R})$ of $G' - A$ of height at least $f_{7.3.1}(k, a)$ whose $\mathfrak{R}$-compass has treewidth at most t and does not intersect $\phi'(S')$.*
Output: *A vertex $v \in V(G) \setminus S'$ such that (G, S', H'_2, ϕ', k) and $(G - v, S', H'_2, \phi', k)$ are equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$.*

Moreover, $f_{7.3.1}(k, a) = \mathcal{O}_{a,\ell_{\mathcal{F}}}(k^c)$, where $c := g_{4.6.12}(a, g_{4.6.11}(a, \ell_{\mathcal{F}})) = \mathcal{O}_{a,\ell_{\mathcal{F}}}(1)$, and the algorithm runs in time $2^{\mathcal{O}_{a,\ell_{\mathcal{F}}}(k \log k + t \log t)} \cdot (n + m)$.

Here is a result with a better dependence on k, and a better running time that we will be able to apply in the bounded genus case. In this case, we do not ask for our flat wall to have bounded treewidth, but to have a planar embedding instead. The proof is deferred to Subsection 7.4.3. Note that here, instead of a single vertex v, we might sometimes find an entire planar block of vertices V that is irrelevant.

Theorem 7.3.2. *Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{F}$ be the collection of obstructions of the graphs embeddable in a surface of genus at most g. There exist a function $f_{7.3.2} : \mathbb{N} \to \mathbb{N}$, whose images are odd integers, and an algorithm with the following specifications:*

`Planar-Irrelevant-Vertex`$(G, S', H'_2, \phi', k, W, \mathfrak{R})$
Input: *An integer $k \in \mathbb{N}$, a graph G, a set $S' \subseteq V(G)$ of size at most k, $(H'_2, \phi') \in \mathcal{L}(G[S'])$, and a flatness pair $(W, \mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi))$ of $G^{S'}_{(H'_2,\phi')}$ of height at least $f_{7.3.2}(k)$ whose $\mathfrak{R}$-compass does not intersect $\phi'(S')$ and is embeddable in a disk with $X \cap Y$ on the boundary.*
Output: *A non-empty set $Y \subseteq V(G) \setminus S'$ such that (G, S', H'_2, ϕ', k) and $(G - Y, S', H'_2, \phi', k)$ are equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$.*

Moreover, $f_{7.3.2}(k) = \mathcal{O}(k)$ and the algorithm runs in time $\mathcal{O}(n + m)$.

The next result essentially states that a part of the solution S can be found in a set A of size $a_{\mathcal{F}}$ such that each vertex of A is adjacent to many vertices of a big enough wall. This is our "obligatory vertex" method. See Figure 7.3 for an illustration. The proof is deferred to Section 7.5.

Lemma 7.3.3. *Let $\mathcal{F}$ be a finite collection of graphs and $\mathcal{L}$ be a hereditary R-action. There exist three functions $f_{7.3.3}, g_{7.3.3}, h_{7.3.3} : \mathbb{N} \to \mathbb{N}$ such that the following holds.*

Let $k \in \mathbb{N}$. Let G be a graph, $S' \subseteq V(G)$ be a set of size at most k, and $(H'_2, \phi') \in \mathcal{L}(G[S'])$. Suppose that $G' := G^{S'}_{(H'_2,\phi')}$ contains a set $A \subseteq V(G')$ of size at least $a_{\mathcal{F}}$ and that there is a wall W in $G' - A$ of height $f_{7.3.3}(k)$. Suppose also that there is a W-canonical partition $\tilde{\mathcal{Q}}$ of $G' - A$ such that each vertex of A is adjacent to at least $g_{7.3.3}(k)$ many $h_{7.3.3}(k)$-internal bags of $\tilde{\mathcal{Q}}$.

Then, for every solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for (G, S', H'_2, ϕ'), it holds that $A' \neq \emptyset$, where $A' := (S \setminus S') \cap A$, and that $|\phi^+(A')| < |A'|$.

Moreover $f_{7.3.3}(k) = \mathcal{O}_{s_{\mathcal{F}}}(k^2)$, $g_{7.3.3}(k) = \mathcal{O}_{s_{\mathcal{F}}}(k^3)$, and $h_{7.3.3}(k) = \mathcal{O}_{s_{\mathcal{F}}}(k^2)$.

Finally, the dynamic programming algorithm presented in Section 7.6 gives the following for graphs of bounded treewidth.

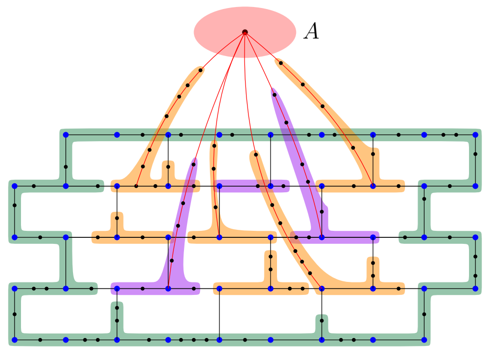

Figure 7.3: Illustration of Lemma 7.3.3.

Theorem 7.3.4. *Let $\mathcal{F}$ be a finite collection of graphs and $\mathcal{L}$ be an R-action. There is an algorithm that, given $k \in \mathbb{N}$, a graph G of treewidth at most w, a set $S' \subseteq V(G)$ of size at most k, and $(H_2', \phi') \in \mathcal{L}(G[S'])$, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2+(k+w)\log(k+w))} \cdot n$ either outputs a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k), or reports a no-instance.*

7.3.2 The general case: proof of Theorem 7.1.5

We now prove our result in the general case. We restate Theorem 7.1.5 for the sake of readability.

Theorem 7.1.5. *Let $\mathcal{F}$ be a finite collection of graphs and let $\mathcal{L}$ be a hereditary R-action. There is an algorithm that, given a graph G, $S' \subseteq V(G)$, $(H_2', \phi') \in \mathcal{L}(G[S'])$, and $k \in \mathbb{N}$, runs in time $2^{\mathsf{poly}_{\mathcal{F}}(k)} \cdot n^2$ and either outputs a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k) or reports a no-instance. Moreover, $\mathsf{poly}_{\mathcal{F}}$ is a polynomial whose degree depends on the maximum detail of a graph in $\mathcal{F}$.*

Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{F}$ be a finite collection of graphs. Let G be a graph, $S' \subseteq V(G)$, $(H_2', \phi') \in \mathcal{L}(G[S'])$, and $k \in \mathbb{N}$. Let us describe here how to solve $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ on (G, S', H_2', ϕ', k).

We set $G' := G^{S'}_{(H_2',\phi')}$ and define the following constants, where $c = g_{4.6.12}(a+b, g_{4.6.11}(a, \ell_{\mathcal{F}})) = \mathcal{O}_{\ell_{\mathcal{F}}}(1)$.

$$
\begin{aligned}
a &= g_{4.6.2}(s_{\mathcal{F}} + a_{\mathcal{F}} - 1) = \mathcal{O}_{\ell_{\mathcal{F}}}(1), & b &= g_{4.6.2}(s_{\mathcal{F}}) = \mathcal{O}_{\ell_{\mathcal{F}}}(1),\\
q &= g_{7.3.3}(k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^3), & p &= h_{7.3.3}(k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^2),\\
l &= (q-1)\cdot(k+b) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^4), & r_5 &= f_{7.3.1}(k, a+b) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^c),\\
t &= f_{4.6.3}(s_{\mathcal{F}}) \cdot r_5 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^c), & r_4 &= \mathsf{odd}(t+3) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^c),\\
r_3 &= f_{4.6.8}(a_{\mathcal{F}} + k, r_4, 1) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+\frac{1}{2}}), & r_2 &= 2 + f_{4.6.2}(s_{\mathcal{F}} + a_{\mathcal{F}} - 1)\cdot r_3 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+\frac{1}{2}}),\\
r_2' &= \mathsf{odd}(\max\{f_{7.3.3}(k), f_{4.6.8}(l+1, r_2, p)\}) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+\frac{5}{2}}), & r_1 &= \mathsf{odd}(f_{4.6.2}(s_{\mathcal{F}})\cdot r_2' + k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+\frac{5}{2}}).
\end{aligned}
$$

Observe that the yes-instances of $\mathcal{L}$-R-$\mathsf{exc}(\mathcal{F})$ exclude $K_{s_{\mathcal{F}}+k}$ as a minor by Observation 7.1.1. Thus, by Proposition 4.2.1, we can always assume that the input graph G has $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$ edges, since otherwise we can directly conclude that (G, k) is a no-instance for $\mathcal{L}$-R-$\mathsf{exc}(\mathcal{F})$.

Given that the algorithm is rather convoluted, we split it into three parts. In the initial steps (Steps 1 and 2), we either find a big enough wall or conclude. Then we analyze what happens when (G, S', H_2', ϕ', k) is a yes-instance of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ containing a big enough wall. That is, we prove that after Step 3, in case of a yes-instance, we either find a flat wall whose compass has bounded treewidth in which case we find an irrelevant vertex in Step 4, or we go to Step 5 and find an apex set intersecting any solution, on which we can branch. Hence, we can apply the final steps (Step 3 to 5), where we either recurse or output a no-instance.

Initial steps

Step 1 (basic check). If $|S'| > k$, we can safely report a no-instance. Hence, we assume in what follows that $|S'| \leq k$.

Step 2 (finding a wall). We run the algorithm `Find-Wall` from Proposition 7.1.2 with input (G, r_1, k) and, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r_1^2+(k+r_1)\log(k+r_1))} \cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2(c+5/2)})} \cdot n$, we either

- conclude that (G, k) is a no-instance of Vertex Deletion to $\mathsf{exc}(\mathcal{F})$, and thus, by Observation 7.1.1, that (G, S', H_2', ϕ', k) is a no-instance of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$, or
- conclude that $\mathsf{tw}(G) \leq f_{7.1.2}(s_{\mathcal{F}}) \cdot r_1 + k$ and solve $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ on (G, S', H_2', ϕ', k) in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2+(r_1+k)\log(r_1+k))} \cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+5/2} \cdot \log k)} \cdot n$ using the algorithm of Theorem 7.3.4, or
- obtain an r_1-wall W_1 of G.

Since we conclude in the first two cases above, we assume henceforth that we have found a r_1-wall W_1 of G.

Interlude: what happens when (G, S', H_2', ϕ', k) is a yes-instance

Given a solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ of the instance (G, S', H_2', ϕ', k), if it exists, let us set $S_r := S \setminus S'$. Note that $G' - S_r$ is a subgraph of $G^S_{(H_2,\phi)}$ and thus belongs to $\mathsf{exc}(\mathcal{F})$.

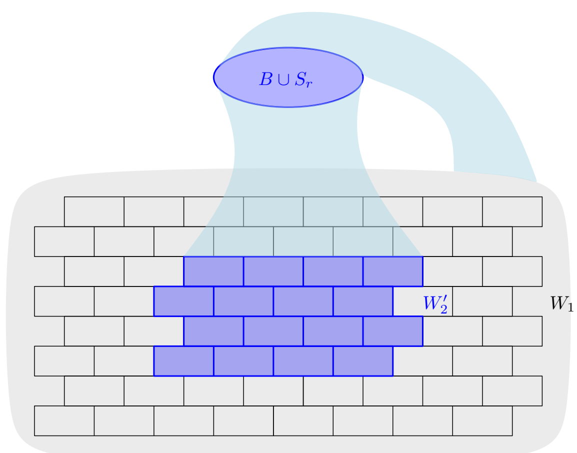

Figure 7.4: $(W_2', \mathfrak{R}_2')$ is a flatness pair of $G' - (S_r \cup B)$.

Claim 7.3.5. *If (S, H_2, ϕ) is a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k), then there exists a set $B \subseteq V(G')$, with $|B| \leq b$, and a flatness pair $(W_2', \mathfrak{R}_2')$ of $G' - (S_r \cup B)$ of height r_2' such that W_2' is a $\tilde{W}_2$-tilt of some subwall $\tilde{W}_2$ of W_1.*

Proof of claim. Since $r_1 \geq f_{4.6.2}(s_{\mathcal{F}}) \cdot r_2' + k$, there is an $(f_{4.6.2}(s_{\mathcal{F}}) \cdot r_2')$-subwall of W_1, say W_1^*, that does not contain vertices of S (by removing the at most k rows and columns containing vertices of S). Hence, W_1^* is a wall of $G - S$ and thus of $G' - S_r \in \mathsf{exc}(\mathcal{F})$.

Since $G' - S_r$ does not contain $K_{s_{\mathcal{F}}}$ as a minor, by Proposition 4.6.2 with input $(G' - S_r, r_2', s_{\mathcal{F}}, W_1^*)$, we know that there is a set $B \subseteq V(G')$, with $|B| \leq b$, and a flatness pair $(W_2', \mathfrak{R}_2')$ of $G' - (S_r \cup B)$ of height r_2' such that W_2' is a $\tilde{W}_2$-tilt of some subwall $\tilde{W}_2$ of W_1^*. ⋄

Let $(W_2', \mathfrak{R}_2')$ be the flatness pair given by Claim 7.3.5. See Figure 7.4 for an illustration. Let $\mathcal{Q}$ be the canonical partition of W_2'. Let $G'_{\mathcal{Q}}$ be the graph obtained by contracting each bag Q of $\mathcal{Q}$ to a single vertex v_Q, and adding a new vertex v_{all} and making it adjacent to each v_Q such that Q is an internal bag of $\mathcal{Q}$. Let $\tilde{A}$ be the set of vertices y of $G' - V(W_2')$ such that there are q internally vertex-disjoint paths from v_{all} to y in $G'_{\mathcal{Q}}$.

Note, as we will use it in Step 5, that, if $\mathcal{Q}'$ is the canonical partition of $\tilde{W}_2$, then $\tilde{A}$ is also the set of vertices y of $G' - V(\tilde{W}_2)$ such that there are q internally vertex-disjoint paths from v_{all} to y in $G'_{\mathcal{Q}'}$.

Claim 7.3.6. $\tilde{A} \subseteq S_r \cup B$.

Proof of claim. To show this, we first prove that, for every $y \in V(G') \setminus (V(W_2') \cup S_r \cup B)$, the maximum number of internally vertex-disjoint paths from v_{all} to y in $G'_{\mathcal{Q}}$ is $k + b + 4$.

Indeed, if y is a vertex in the $\mathfrak{R}_2'$-compass of W_2' (but not a vertex in $V(W_2')$), then there are at most $k + b$ such paths that intersect the set $S_r \cup B$ and at most four paths that do not intersect $S_r \cup B$ (in the graph $G'_{\mathcal{Q}} - (S_r \cup B)$) due to the fact that $(W_2', \mathfrak{R}_2')$ is a flatness pair of $G' - (S_r \cup B)$.

If y is not a vertex in the $\mathfrak{R}_2'$-compass of W_2', then, since by the definition of flatness pairs the perimeter of W_2' together with the set $S_r \cup B$ separate y from the $\mathfrak{R}_2'$-compass of W_2', every collection of internally vertex-disjoint paths from v_{all} to y in $G'_{\mathcal{Q}}$ should intersect the set $\{v_{Q_{\text{ext}}}\} \cup S_r \cup B$, where Q_{ext} is the external bag of $\mathcal{Q}$.

Therefore, in both cases, the maximum number of internally vertex-disjoint paths from v_{all} to y in G' is $k + b + 4$. Since $k + b + 4 < q$, we have that $y \notin \tilde{A}$. Hence, given that $\tilde{A} \subseteq V(G') \setminus V(W_2')$, we conclude that $\tilde{A} \subseteq S_r \cup B$. ⋄

Given a W_2'-canonical partition $\tilde{\mathcal{Q}}$ of $G' - (S_r \cup B)$, we set $A_{\tilde{\mathcal{Q}}}$ to be the set of vertices in $S_r \cup B$ that are adjacent to vertices of at least q p-internal bags of $\tilde{\mathcal{Q}}$. Note that $A_{\tilde{\mathcal{Q}}} \subseteq \tilde{A}$ and therefore $|A_{\tilde{\mathcal{Q}}}| \leq |\tilde{A}|$. Remember that $\tilde{\mathcal{Q}}$ is obtained by enhancing $\mathcal{Q}$ on $G' - (S_r \cup B)$ and is not unique.

Claim 7.3.7. *If there is a W_2'-canonical partition $\tilde{\mathcal{Q}}$ of $G' - (S_r \cup B)$ such that $|A_{\tilde{\mathcal{Q}}}| < a_{\mathcal{F}}$, then*

- *(a) there is an r_2-subwall W_2 of W_1 such that the algorithm* `Grasped-or-Flat` *of Proposition 4.6.2 with input $(D_{W_2}, r_3, s_{\mathcal{F}} + a_{\mathcal{F}} - 1, W_2^*)$ outputs a set $A \subseteq V(D_{W_2})$ with $|A| \leq a$ and a flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} - A$ of height r_3, such that W_3 is a tilt of some subwall $\tilde{W}_3$ of W_2, where*
  - *W_2^* is the central $(r_2 - 2)$-subwall of W_2 and*
  - *D_{W_2} is the graph obtained from G' after removing the perimeter of W_2 and taking the connected component containing W_2^*, and*
- *(b) the algorithm* `Clique-or-twFlat` *of Proposition 4.6.3 with input $(D_{W_4}, r_5, s_{\mathcal{F}})$ outputs a set A' of size at most b and a regular flatness pair $(W_5, \mathfrak{R}_5)$ of $D_{W_4} - A'$ of height r_5 whose $\mathfrak{R}_5$-compass has treewidth at most t and does not intersect $\phi'(S')$, where*
  - *W_4 is a wall in the collection $\mathcal{W} = \{W^1, \dots, W^{a_{\mathcal{F}}+k}\}$,*

– *W_4^* is the central $(r_4 - 2)$-subwall of W_4, and*

– *D_{W_4} is the graph obtained from D_{W_2} after removing A and the perimeter of W_4 and taking the connected component containing W_4^*.*

Proof of claim. Given that $|A_{\tilde{\mathcal{Q}}}| < a_{\mathcal{F}}$, at most $a_{\mathcal{F}} - 1$ vertices of $S_r \cup B$ are adjacent to vertices of at least q p-internal bags of $\tilde{\mathcal{Q}}$. This means that the p-internal bags of $\tilde{\mathcal{Q}}$ that contain vertices adjacent to some vertex of $(S_r \cup B) \setminus A_{\tilde{\mathcal{Q}}}$ are at most $(q-1) \cdot (k+b) = l$.

Given that $r_2' \geq f_{4.6.8}(l+1, r_2, p)$, there is a collection $\mathcal{W} = \{W^1, \ldots, W^{l+1}\}$ of $l+1$ r_2-subwalls of W_2' in G' respecting the properties of the output of the algorithm `Packing` of Proposition 4.6.8 with input $(l+1, r_2, p, G', W_2', \mathfrak{R}_2')$. The fact that the p-internal bags of $\tilde{\mathcal{Q}}$ that contain vertices adjacent to some vertex of $(S_r \cup B) \setminus A_{\tilde{\mathcal{Q}}}$ are at most l implies that there exists an $i \in [l+1]$ such that no vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2'}(W^i))$ is adjacent, in G', to a vertex in $(S_r \cup B) \setminus A_{\tilde{\mathcal{Q}}}$. Let W_2 be the subwall of W_1 such that W^i is a tilt of W_2. It exists given that W^i is a subwall of W_2', that is a tilt of some subwall W' of W_1. Remember that W_2^* is the central $(r_2 - 2)$-subwall of W_2, which is also the central $(r_2 - 2)$-subwall of W^i, and that D_{W_2} is the graph obtained from G' by removing the perimeter of W_2 and taking the connected component that contains W_2^*.

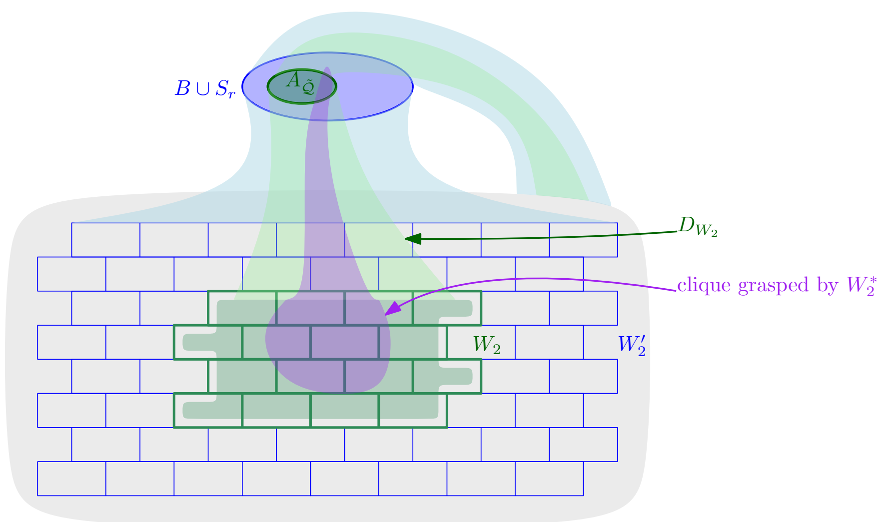

Figure 7.5: $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2'}(W_2))$ is not adjacent to any vertex in $(S_r \cup B) \setminus A_{\tilde{\mathcal{Q}}}$.

Since no vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2'}(W^i))$ is adjacent, in G', to a vertex in $(S_r \cup B) \setminus A_{\tilde{\mathcal{Q}}}$, any path in D_{W_2} going from a vertex of W_2^* to a vertex in S_r must intersect a vertex of $A_{\tilde{\mathcal{Q}}}$. Thus, there is no model of $K_{s_{\mathcal{F}}+a_{\mathcal{F}}-1}$ grasped by W_2^* in D_{W_2}, because otherwise, $K_{s_{\mathcal{F}}}$ would be a minor of the connected component of $D_{W_2} - A_{\tilde{\mathcal{Q}}}$ containing W_2^*, and thus of $G' - S_r$. See Figure 7.5 for an illustration. So, by applying the algorithm `Grasped-or-Flat` of Proposition 4.6.2 with input $(D_{W_2}, r_3, s_{\mathcal{F}} + a_{\mathcal{F}} - 1, W_2^*)$, since $r_2 - 2 \geq f_{4.6.2}(s_{\mathcal{F}} + a_{\mathcal{F}} - 1) \cdot r_3$, we should find a set $A \subseteq V(D_{W_2})$ with $|A| \leq a$ and a flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} - A$ of height r_3, such that W_3 is a tilt of some subwall $\tilde{W}_3$ of W_2.

Let $\tilde{\mathcal{Q}}'$ be a W_3-canonical partition of $D_{W_2} - A$. Given that $r_3 \geq f_{4.6.8}(a_{\mathcal{F}} + k, r_4, 1)$, there is a collection $\mathcal{W}' = \{W^1, \ldots, W^{a_{\mathcal{F}}+k}\}$ of r_4-subwalls of W_3 respecting the properties of the output of the algorithm `Packing` of Proposition 4.6.8 with input $(a_{\mathcal{F}} + k, r_4, 1, D_{W_2} - A, W_3, \mathfrak{R}_3)$. Since $|A_{\tilde{\mathcal{Q}}}| < a_{\mathcal{F}}$ and $|(\phi')^+(S')| \leq |S'| \leq k$, there is an $i \in [a_{\mathcal{F}} + k]$ such that $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W^i))$ does not intersect $A_{\tilde{\mathcal{Q}}}$ nor $\phi'(S')$. See Figure 7.6 for an illustration.

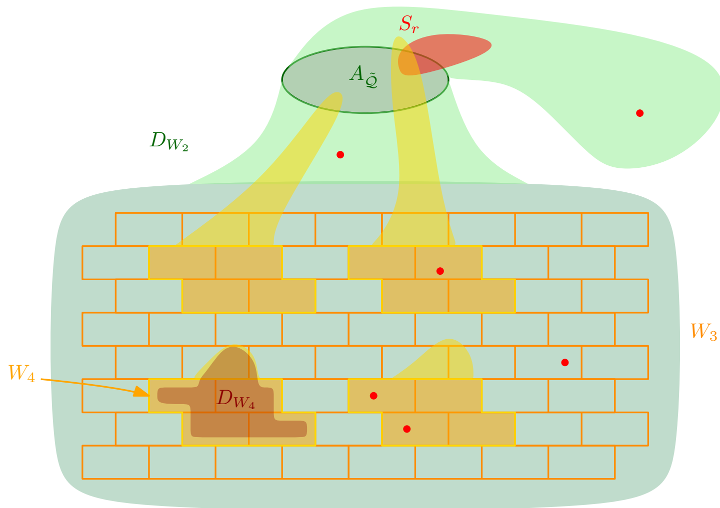

Figure 7.6: $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W_4))$ does not contain any vertex from $\phi'(S')$ (red vertices), nor from $A_{\tilde{Q}}$ (and thus from S_r). The small walls in orange are the walls of $\mathcal{W}'$, and their influence is represented in yellow.

Let $W_4 := W^i$. Remember that W_4^* is the central (r_4-2)-subwall of W_4 and that D_{W_4} is the graph obtained from D_{W_2} after removing A and the perimeter of W_4 and taking the connected component containing W_4^*. Observe that any path between a vertex of S_r and a vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W_4))$ in D_{W_2} intersects $A_{\tilde{Q}}$. Since $A_{\tilde{Q}}$ does not intersect $V(\bigcup \mathsf{influence}_{\mathfrak{R}_3}(W_4))$, it implies that $A_{\tilde{Q}}$ does not intersect D_{W_4}, and thus $S_r \cap D_{W_4} = \emptyset$. Therefore, D_{W_4} is a subgraph of $G' - S_r$ and $K_{s_\mathcal{F}}$ is not a minor of D_{W_4}. Moreover, W_4^* is a wall of D_{W_4} of height $r_4 - 2 \geq t+1$, so $\mathsf{tw}(D_{W_4}) > t = f_{4.6.3}(s_\mathcal{F}) \cdot r_5$. Therefore, by applying the algorithm `Clique-or-twFlat` of Proposition 4.6.3 with input $(D_{W_4}, r_5, s_\mathcal{F})$, we should obtain a set A' of size at most b and a regular flatness pair $(W_5, \mathfrak{R}_5)$ of $D_{W_4} - A'$ of height r_5 whose $\mathfrak{R}_5$-compass has treewidth at most t. $\diamond$

Final steps

Step 3a (finding a flat wall). We consider all the $\binom{r_1}{r_2}^2 = 2^{\mathcal{O}_{\ell_\mathcal{F}}(k^{c+1/2}\log k)}$ r_2-subwalls of W_1 not containing a vertex of S' (so that they are walls of both G and G'). For each one of them, say W_2, let W_2^* be the central (r_2-2)-subwall of W_2 and let D_{W_2} be the graph obtained from G' after removing the perimeter of W_2 and taking the connected component containing W_2^*. Given that $r_2 - 2 = f_{4.6.2}(s_\mathcal{F} + a_\mathcal{F} - 1) \cdot r_3$, we can apply the algorithm `Grasped-or-Flat` of Proposition 4.6.2 with input $(D_{W_2}, r_3, s_\mathcal{F} + a_\mathcal{F} - 1, W_2^*)$. This can be done in time $\mathcal{O}_{s_\mathcal{F}}(k\sqrt{\log k} \cdot n)$. If, for some of these subwalls, the result is a set $A \subseteq V(D_{W_2})$ with $|A| \leq a$ and a flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} - A$ of height r_3 then we proceed to **Step 3b** for each such a subwall. Otherwise, we proceed to **Step 5**.

Step 3b (finding a flat wall whose compass has bounded treewidth). Given that $r_3 = f_{4.6.8}(a_\mathcal{F} + k, r_4, 1)$, we can apply the algorithm `Packing` of Proposition 4.6.8 with input $(a_\mathcal{F} + k, r_4, 1, D_{W_2} - A, W_3, \mathfrak{R}_3)$ to compute in time $\mathcal{O}_{s_\mathcal{F}}(k\sqrt{\log k} \cdot n)$ a collection $\mathcal{W} = \{W^1, \ldots, W^{a_\mathcal{F}+k-1}\}$ of r_4-subwalls of W_3 that respects the properties of the output of Proposition 4.6.8.

For $i \in [a_\mathcal{F} + k - 1]$, let W^{i*} be the central $(r_4 - 2)$-subwall of W^i and let D_{W^i} be the graph obtained from D_{W_2} after removing A and the perimeter of W^i and taking the connected component containing W^{i*}. Run the algorithm `Clique-or-twFlat` of Proposition 4.6.3 with input $(D_{W^i}, r_5, s_\mathcal{F})$. This takes time $2^{\mathcal{O}_{\ell_\mathcal{F}}(r_5^2)} \cdot n = 2^{\mathcal{O}_{\ell_\mathcal{F}}(k^{2c})} \cdot n$. If for one of these subwalls the result is a set A' of size at most b and a regular flatness pair $(W_5, \mathfrak{R}_5)$ of $D_{W^i} - A'$ of height r_5 whose $\mathfrak{R}_5$-compass has treewidth at most t and does not intersect S', then we set $W_4 := W^i$ and we proceed to **Step 4**.

If, for every subwall W_2, we did not find such a pair $(W_5, \mathfrak{R}_5)$, then we proceed to **Step 5**.

Step 4 (irrelevant vertex case). Let $\mathfrak{R}'_5$ be the 5-tuple obtained by adding all vertices of $G' - V(D_{W_4}) - A$ to the set in the first coordinate of $\mathfrak{R}_5$.

Claim 7.3.8. *$(W_5, \mathfrak{R}'_5)$ is a regular flatness pair of $G' - (A \cup A')$ whose $\mathfrak{R}'_5$-compass has treewidth at most t and does not intersect $\phi'(S')$.*

Proof of claim. Remember that, given a wall W, $D(W)$ is the perimeter of W.

$(W_5, \mathfrak{R}_5)$ is a flatness pair of $D_{W_4} - A'$. By the definition of D_{W_4}, the vertices of $D_{W_2} - V(D_{W_4}) - A - D(W_4)$ are only adjacent to $D(W_4)$ and A in D_{W_2}. Therefore, $(W_5, \mathfrak{R}''_5)$ is a flatness pair of $D_{W_2} - (A \cup A')$, where $\mathfrak{R}''_5$ is the 5-tuple obtained by adding all vertices of $D_{W_2} - V(D_{W_4}) - A$ to the set in the first coordinate of $\mathfrak{R}_5$.

Also, by the definition of D_{W_2}, the vertices of $G' - V(D_{W_2}) - D(W_2)$ are only adjacent to $D(W_2)$ in G'. Therefore, $(W_5, \mathfrak{R}'_5)$ is a flatness pair of $G' - (A \cup A')$, where $\mathfrak{R}'_5$ is the 5-tuple obtained by adding all vertices of $G' - D_{W_2}$ to the set in the first coordinate of $\mathfrak{R}''_5$. Therefore, $\mathfrak{R}'_5$ is indeed the 5-tuple obtained by adding all vertices of $G' - V(D_{W_4}) - A$ to the set in the first coordinate of $\mathfrak{R}_5$.

Given that $\mathsf{Compass}_{\mathfrak{R}_5}(W_5) = \mathsf{Compass}_{\mathfrak{R}'_5}(W_5)$ and that $(W_5, \mathfrak{R}_5)$ is regular with a $\mathfrak{R}_5$-compass of treewidth at most t that does not intersect $\phi'(S')$, this is also the case for $(W_5, \mathfrak{R}'_5)$. Hence the result. $\diamond$

Given that $r_5 = f_{7.3.1}(k, a+b)$, we can apply the algorithm `Irrelevant-Vertex` of Theorem 7.3.1 with input $(G, S', H'_2, \phi', k, A \cup A', a+b, W_5, \mathfrak{R}'_5, t)$, which outputs, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(t \log t + k \log k)} \cdot (n+m) = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^c \log k)} \cdot n$, a vertex v such that (G, S', H'_2, ϕ', k) and $(G - v, S', H'_2, \phi', k)$ are equivalent instances of $\mathcal{L}$-R-$\mathsf{exc}(\mathcal{F})$. Then the algorithm runs recursively on the equivalent instance $(G - v, S', H'_2, \phi', k)$.

Step 5 (branching case). Consider all the r'_2-subwalls of W_1 that do not contain vertices of S', which are at most $\binom{r_1}{r'_2}^2 = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+5/2} \log k)}$ many, and for each of them, say $\tilde{W}_2$, compute its canonical partition $\mathcal{Q}'$. Note that $\tilde{W}_2$ is a wall of $G - S'$, and thus of G'. Then, in G', we contract each bag Q of $\mathcal{Q}'$ to a single vertex v_Q, and add a new vertex v_{all} and make it adjacent to each v_Q such that Q is an internal bag of $\mathcal{Q}'$. In the resulting graph $G'_{\mathcal{Q}'}$, for every vertex y of $G' - V(\tilde{W}_2)$, check, using augmenting paths from usual maximum flow techniques [87], whether there are q internally vertex-disjoint paths from v_{all} to y in time $\mathcal{O}(q \cdot m) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^4 \sqrt{\log k} \cdot n)$. Let $\tilde{A}$ be the set of all such y's.

Note that, if (G, S', H'_2, ϕ', k) is a yes-instance, then, by Claim 7.3.5 and Claim 7.3.6, there is a $\tilde{W}_2$ such that $|\tilde{A}| \leq k + b$, and by Claim 7.3.7, $|\tilde{A}| \geq a_{\mathcal{F}}$, since otherwise we would have gone to Step 4.

Hence, for each $\tilde{W}_2$ such that $a_{\mathcal{F}} \leq |\tilde{A}| \leq k + b$, we do the following. We consider all the $\binom{\tilde{A}}{a_{\mathcal{F}}} = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\log k)}$ subsets of $\tilde{A}$ of size $a_{\mathcal{F}}$. For each one of them, say A^*, construct a $\tilde{W}_2$-canonical partition $\tilde{\mathcal{Q}}'$ of $G' - A^*$ by enhancing $\mathcal{Q}'$ on $G' - A^*$, such that we first greedily increase the size of the external bag Q_{ext}. Note that if $\tilde{W}_2$ is the wall of Claim 7.3.5, then there is a W'_2-canonical partition $\tilde{\mathcal{Q}}$ of $G' - (S_r \cup B)$ and a set $A_{\tilde{\mathcal{Q}}} \subseteq \tilde{A}$ such that every bag of $\tilde{\mathcal{Q}}'$ contains exactly one bag of $\tilde{\mathcal{Q}}$. Therefore, by Claim 7.3.7 and given that we did not go to Step 4, we conclude that, if (G, S', H'_2, ϕ', k) is a yes-instance, then there is a set A^* whose vertices are all adjacent to vertices of

q p-internal bags of $\tilde{\mathcal{Q}}'$. Therefore, if, for every $\tilde{W}_2$ such that $a_{\mathcal{F}} \leq |\tilde{A}| \leq k+b$ and for every $A^* \subseteq \tilde{A}$, the vertices of A^* are not all adjacent to vertices of q p-internal bags of $\tilde{\mathcal{Q}}'$, we report a no-instance.

Assume now that we found $\tilde{W}_2$ and A^* such that the vertices of A^* are all adjacent to vertices of q p-internal bags of $\tilde{\mathcal{Q}}'$. Then, given that $r_2' \geq f_{7.3.3}(k)$, $q = g_{7.3.3}(k)$, and $p = h_{7.3.3}(k)$, by Lemma 7.3.3, for every solution (S, H_2, ϕ) of $\mathcal{L}$-AR-exc$(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k), it holds that $(S \setminus S') \cap A^* \neq \emptyset$. Let $S'' := S' \cup (S \cap A^*)$ and (H_2'', ϕ'') be the restriction of (H_2, ϕ) to S''. Hence, we guess (S'', H_2'', ϕ'') and solve the instance $(G, S'', H_2'', \phi'', k)$. Since we add at most $a_{\mathcal{F}}$ vertices to extend (S', H_2', ϕ') to (S'', H_2'', ϕ'') and given that H_2 has at most k vertices, there are at most $2^{a_{\mathcal{F}}}$ choices for S'', at most $2^{a_{\mathcal{F}} \cdot k}$ choices for H_2'', and at most $(k+1)^{a_{\mathcal{F}}}$ choices for ϕ'', which means at most $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k)}$ possible guesses for (S'', H_2'', ϕ''). Therefore, the algorithm runs recursively on $(G, S'', H_2'', \phi'', k)$ for each such (S'', H_2'', ϕ''). If one of them is a yes-instance with solution (S, H_2, ϕ), then (G, S', H_2', ϕ', k) is a yes-instance with the same solution. Otherwise, we report a no-instance.

Running time. Notice that Step 5, when applied, takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+5/2} \log k)} \cdot n^2$, because we apply the flow algorithm to each of the $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c+5/2} \log k)}$ r_2'-subwalls and for each vertex of G. However, the search tree created by the branching technique has at most $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k)}$ branches and depth at most k, since the size of the partial solution strictly increase each time. So Step 5 cannot be applied more than $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2)}$ times during the course of the algorithm. Since Step 1 runs in time $\mathcal{O}_{\ell_{\mathcal{F}}}(1)$, Step 2 runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2(c+5/2)})} \cdot n$, Step 3 in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c})} \cdot n$, and Step 4 in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{c \log c})} \cdot n$, and that they all may be applied at most n times, the claimed time complexity follows: the algorithm runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2(c+5/2)})} \cdot n^2$.

7.3.3 The special case of bounded genus: proof of Theorem 7.1.6

When $\mathcal{H} = \mathsf{exc}(\mathcal{F})$ is the class of graphs embeddable in a surface Σ_g of Euler genus at most g, we can modify the general algorithm so that the degree of k in the running time does not depend on $\mathcal{H}$. This is due to two facts. First, we now have $a_{\mathcal{F}} = 1$, given that, for some t that depends on g, $K_{3,t}$, which has apex number one, does not embed in Σ_g. Hence, when applying Lemma 7.3.3, the obligatory set A contains a unique vertex v. This implies that we do not need to branch on A, but instead, v is an "obligatory vertex". In particular, given that the size of A must strictly decrease after the modification, this further implies that v must be deleted.

Second, by Theorem 7.3.2, we can find an irrelevant vertex inside a flat wall whose height is far smaller than the one required in Theorem 7.3.1 for the general case. This changes the algorithm a bit, because the flat wall we require now needs to have a compass *embeddable in a disk* instead of a compass of *bounded treewidth*. Hence, as in the general case, we find a flat wall (Steps 1, 2, 3a). But, while in the general case we used this flat wall to further find a flat wall W whose compass has bounded treewidth (Step 3b), we now either find an obligatory vertex in Step 4, or proceed to Step 5 where we find a flat subwall W' of W whose compass is in $\mathcal{H}$, in which we argue using the argument of Section 7.3.3 that there is another flat subwall W'' of W' whose compass is embeddable in a disk, where we finally find an irrelevant vertex. If we did not find a flat wall in Step 3a, then, similarly to the branching case (Step 5) in the general setting, we find an obligatory vertex in Step 7.

Finding a disk-embeddable wall

In the algorithm, we will need to prove that, if the compass C of a flat wall is embedded in a surface of bounded Euler genus, then C contains a smaller flat wall that is embeddable in a disk such that its perimeter is on the boundary of the disk. Namely, we need this property to apply Lemma 7.4.1

when finding an irrelevant vertex. To do so, we use the following result from Demaine, Hajiaghayi, and Thilikos [86].

Proposition 7.3.9 (Lemma 4.7 in [86])**.** *Let G be a graph $(\emptyset, \emptyset)$-embeddable in a surface Σ of Euler genus g and assume that $\mathsf{tw}(G) \geq 4(r-12g)(g+1)$. Then there exists some $(r-12g, g)$-gridoid H, $(\emptyset, F)$-embeddable in $\mathbb{S}_0$ for some $F \subseteq E(H)$ with $|F| \leq g$, such that there exists some contraction mapping from G to H with respect to their corresponding embeddings.*

Let us briefly introduce the undefined terms used in the above statement. A graph G is (S, F)-*embeddable* in a surface Σ, where $S \subseteq V(G)$, $F \subseteq E(G)$, and F is a superset of the edges with an endvertex in S, if the graph $G^- := (V(G) \setminus S, E(G) \setminus F)$ admits a *2-cell* embedding in Σ, that is, an embedding in which every face is homeomorphic to an open disk. For two positive integers r, k, a graph G is an (r, k)-*gridoid* if it is (S, F)-embeddable in $\mathbb{S}_0$ for some pair S, F, where $|F| \leq k$, $F(G[S]) = \emptyset$, and G^- is a *partially triangulated* $(r' \times r')$-grid embedded in $\mathbb{S}_0$ for some $r' \geq r$, that is, a graph obtained from an $(r' \times r')$-grid by adding some chords to some of its faces. Finally, without entering into unnecessary technical details, a *contraction mapping* is a strengthening of a graph being a contraction of another graph that preserves some aspects of the embedding in a surface during the contractions. In a nutshell, the statement of Proposition 7.3.9 should be interpreted as H occurring as a contraction in G in such a way that H can be embedded "nicely inside the original embedding of G", in particular the preimages of the faces of H via the contraction mapping are faces of G; see [86, Section 3.1] for more details.

In our setting, our goal by using Proposition 7.3.9 is to guarantee that, in **Step 5** below, given a large flat wall embedded in a surface of bounded genus, it is possible to find inside it a still large flat *plane* subwall that is *nicely embedded* in the original one. To this end, it is enough to argue that, once we have at hand the gridoid H given by Proposition 7.3.9, we can find a large piece of it that is $(\emptyset, \emptyset)$-embeddable in $\mathbb{S}_0$.

Note that, in order to apply Proposition 7.3.9 in our algorithm and obtain an overall quadratic time, we need to obtain the embedded gridoid H in *linear time*, so that an irrelevant vertex can indeed be found in linear time. Let us briefly sketch how this can be done. The proof of [86, Lemma 4.7] proceeds by induction on the Euler genus g, the base case following by the planar exclusion theorem of Robertson and Seymour [282]. It then distinguishes two cases according to whether the *representativity* of G (also called *face-width* in the literature) it at least $\ell := 4(r-12g)$ or not. This can be decided in time $\mathcal{O}(g\ell n)$ by [46, Theorem 10], where $n = |V(G)|$. If this is indeed the case, applying [81, Lemma 3.3] yields the desired output. The proof of [81, Lemma 3.3] consists in simple local operations based on a notion of distance that uses the *existence* of an object called *respectful tangle*, proved to exist in [269, Theorem 4.1]. It is easy to verify that these local operations and the definition of the distance function can be done in linear time.

Otherwise, if the representativity is less than ℓ, the proof first reduces the Euler genus of the surface by applying a so-called *splitting* operation, and then uses the induction hypothesis and a sequence of local operations that identify a set of edges to be contracted to obtain the desired output. Again, these local operations are easily seen to be done in linear time.

The algorithm

We now have all the necessary ingredients to prove the algorithm. Before proceeding to the proof of Theorem 7.1.6, we restate it for the sake of readability.

Theorem 7.1.6. *Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{H}$ be the class of graphs embeddable in a surface Σ of Euler genus at most g. There is an algorithm that, given a graph G, $S' \subseteq V(G)$,*

$(H_2', \phi') \in \mathcal{L}(G[S'])$, *and* $k \in \mathbb{N}$, *runs in time* $2^{\mathcal{O}_g(k^9)} \cdot n^2$ *and either outputs a solution of* $\mathcal{L}$-AR-$\mathcal{H}$ *for the instance* (G, S', H_2', ϕ', k) *or reports a* no-*instance.*

Let $\mathcal{L}$ be a hereditary R-action. Let G be a graph, $S' \subseteq V(G)$, $(H_2', \phi') \in \mathcal{L}(G[S'])$, and $k \in \mathbb{N}$. Let us describe in what follows how to solve $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ on (G, S', H_2', ϕ', k).

We set $G' := G^{S'}_{(H_2', \phi')}$ and define the following constants. Remember that $a_{\mathcal{F}} = 1$ in this section.

$$\begin{aligned}
a, b &= g_{4.6.2}(s_{\mathcal{F}}) = \mathcal{O}_{\ell_{\mathcal{F}}}(1), & q &= g_{7.3.3}(k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^3),\\
l &= (q-1) \cdot (k+b) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^4), & p &= h_{7.3.3}(k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^2),\\
d &= (q-1) \cdot a + k + 1 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^3), & r_6 &= f_{7.3.2}(k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k),\\
r_5 &= \mathsf{odd}(\max\{12g, (2g+2) \cdot 2r_6\}) = \mathcal{O}_{\ell_{\mathcal{F}}}(k), & r_4 &= 4r_5 \cdot (g+1) + 1 = \mathcal{O}_{\ell_{\mathcal{F}}}(k),\\
r_3 &= f_{4.6.8}(d, r_4, 1) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{5/2}), & r_2 &= 2 + f_{4.6.2}(s_{\mathcal{F}}) \cdot r_3 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{5/2}),\\
r_2' &= \mathsf{odd}(\max\{f_{7.3.3}(k), f_{4.6.8}(l+1, r_2, p)\}) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{9/2}), & r_1 &= \mathsf{odd}(f_{4.6.2}(s_{\mathcal{F}}) \cdot r_2' + k) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{9/2}).
\end{aligned}$$

Recall that we assume that G has $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$ edges.

Step 1, 2, and 3a are done as in the general case. If, for some of the r_2-subwalls of W_1 not containing a vertex of S', we find a set $A \subseteq V(D_{W_2})$ with $|A| \leq a$ and a flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} - A$ of height r_3 then we go to **Step 4**. Otherwise, we go to **Step 7**.

Step 4 (obligatory vertex case 1). Let $\mathfrak{R}_3'$ be the 5-tuple obtained by adding all the vertices of $G' - V(D_{W_2}) - A$ in the set in the first coordinate of $\mathfrak{R}_3$. Similarly to Claim 7.3.8, we get that $(W_3, \mathfrak{R}_3')$ is a flatness pair of $G' - A$. We apply the algorithm of Proposition 4.6.7 to find in time $\mathcal{O}(k\sqrt{\log k} \cdot n)$ a regular flatness pair $(W_3^*, \mathfrak{R}_3^*)$ of $G' - A$ of height r_3 such that $\mathsf{Compass}_{\mathfrak{R}_3^*}(W_3^*) \subseteq \mathsf{Compass}_{\mathfrak{R}_3'}(W_3)$.

Let $\tilde{\mathcal{Q}}$ be a W_3^*-canonical partition of $G' - A$. If there is a vertex $v \in A$ that has neighbors in at least q p-internal bags of $\tilde{\mathcal{Q}}$, then, given that $r_2' \geq f_{7.3.3}(k)$, $q = g_{7.3.3}(k)$, and $p = h_{7.3.3}(k)$, by Lemma 7.3.3, for every solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathcal{P}$ for the instance (G, S', H_2', ϕ', k), it holds that $v \in S \setminus S'$ and that $|\phi^+(v)| < |v| = 1$. In other words, $\phi(v) = \emptyset$, that is, v must be deleted. Let $S'' := S' \cup \{v\}$ and $\phi'' := \phi' \cup (v \mapsto \emptyset)$. Hence, if $(H_2', \phi'') \in \mathcal{L}(G[S''])$, then (G, S, H_2', ϕ', k) and (G, S'', H_2, ϕ'', k) are equivalent instances of $\mathcal{L}$-AR-$\mathcal{P}$. Hence, the algorithm runs recursively on (G, S'', H_2, ϕ'', k) and outputs its result. Otherwise, if $(H_2', \phi'') \notin \mathcal{L}(G[S''])$, then we report a no-instance. Thus, we can now assume that every vertex of A has neighbors in at most $q - 1$ p-internal bags of $\tilde{\mathcal{Q}}$ and go to **Step 5**.

Step 5 (finding a flat wall whose compass is disk-embeddable). Given that $r_3 = f_{4.6.8}(d, r_4, 1)$, we apply the algorithm `Packing` of Proposition 4.6.8 with input $(d, r_4, 1, D_{W_2} - A, W_3, \mathfrak{R}_3)$ to find in time $\mathcal{O}(k\sqrt{\log k} \cdot n)$ a collection $\mathcal{W} = \{W^1, \dots, W^d\}$ of r_4-subwalls of W_3^* respecting the properties of the output of Proposition 4.6.8. For $i \in [d]$, we apply the algorithm of Proposition 4.6.6 to find in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$ a W^i-tilt $(\tilde{W}^i, \tilde{\mathfrak{R}}_i)$ of $(W_3^*, \mathfrak{R}_3^*)$.

Given that each vertex of A has neighbors in at most $q-1$ p-internal bags of $\tilde{\mathcal{Q}}$ and that $d \geq (q-1) \cdot a + k + 1$, by the pigeonhole principle, there is $I \subseteq [d]$ of size at least $k+1$ such that, for each $i \in I$, $\mathsf{Compass}_{\tilde{\mathfrak{R}}_i}(\tilde{W}^i)$ has no neighbors in A. Note that, by the properties of Proposition 4.6.8, the graphs $\mathsf{Compass}_{\tilde{\mathfrak{R}}_i}(\tilde{W}^i)$ are pairwise disjoint for $i \in I$. Let $\tilde{\mathfrak{R}}_i'$ be the 5-tuple obtained by adding all the vertices of A in the set in the first coordinate of $\tilde{\mathfrak{R}}_i$. Therefore, for each $i \in I$, $(\tilde{W}^i, \tilde{\mathfrak{R}}_i')$ is a flatness pair of G'.

For each $i \in I$, let $C_i := \mathsf{Compass}_{\tilde{\mathfrak{R}}'_i}(\tilde{W}^i)$ and let C_i^+ be the graph obtained from C_i by adding a vertex v_i adjacent to each vertex of $X_i \cap Y_i$, where X_i (resp. Y_i) is the first (resp. second) coordinate of $\tilde{\mathfrak{R}}_i$. For each $i \in I$, we check whether C_i^+ embeds in Σ_g, and if it does, we find such an embedding. This can be done in linear time by using the algorithm of Mohar [232]. Given that $|I| \geq k+1$, for any solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathcal{P}$ for the instance (G, S', H_2', ϕ', k), there is $j \in I$ such that C_j does not contain a vertex of $\phi(S)$ (and in particular of $\phi'(S')$). Therefore, C_j^+ must have Euler genus at most g. Thus, if, for each $i \in I$, C_i^+ is not embeddable in Σ_g, we report a no-instance. Otherwise, there is $j \in I$ such that C_j^+ is embeddable in Σ_g and does not contain a vertex of $\phi'(S')$. Then, by Proposition 7.3.9 and the discussion after it, given that $\mathsf{tw}(C_j^+) \geq r_4 \geq 4r_5 \cdot (g+1)$ and that $r_5 \geq 12g$, we can find in linear time an edge set $F \subseteq E(C_j^+)$ of size at most g and an (r_5, g)-gridoid H that is $(\emptyset, F)$-embeddable in $\mathbb{S}_0$, such that there exists a contraction mapping from C_j^+ to H with respect to their corresponding embeddings. Let M be the union of v_j and the set of vertices of H incident to edges in F. We have $|M| \leq 2g+1$. Given that $r_5 \geq (2g+2) \cdot 2r_6$, again by the pigeonhole principle there is an induced subgraph of $H - M$ that is a partially triangulated $(2r_6 \times 2r_6)$-grid Γ. In particular, there is a contraction mapping from C_j^+ to Γ with respect to their corresponding embeddings, with Γ embedded in $\mathbb{S}_0$. Given that Γ is 3-connected (if we dissolve the corners), by Whitney's theorem [310], it has a unique embedding in $\mathbb{S}_0$. In particular, the perimeter of Γ bounds a face. Also, there is an elementary r_6-wall W_6 that is a subgraph of Γ. Given that the maximum degree in W_6 is three and that it avoids the vertices in M, it implies that C_j^+, and thus G', contains a r_6-wall W_6' as a subgraph (whose contraction gives W_6). Hence, given that the contractions preserve the embedding, we conclude that there is some rendition $\mathfrak{R}_6$ such that $(W_6', \mathfrak{R}_6)$ is a flatness pair of G' whose $\mathfrak{R}_6$-compass does not contain a vertex of $\phi'(S')$ and is embeddable in a disk with $X_6 \cap Y_6$ on its boundary, where X_6 (resp. Y_6) is the first (resp. second) coordinate of $\tilde{\mathfrak{R}}_6$.

Step 6 (irrelevant vertex case). Thus, given that $r_6 = f_{7.3.2}(k)$, we can apply the algorithm `Planar-Irrelevant-Vertex` of Theorem 7.3.2 with input $(G, S', H_2', \phi', k, W_6', \mathfrak{R}_6)$. It outputs in time $\mathcal{O}(k\sqrt{\log k} \cdot n)$ a non-empty set $Y \subseteq V(G) \setminus S'$ such that (G, S', H_2', ϕ') and $(G - Y, S', H_2', \phi')$ are equivalent instances of $\mathcal{L}$-AR-$\mathcal{P}$. Hence, the algorithm runs recursively for the instance $(G - Y, S', H_2', \phi')$ and concludes.

Step 7 (obligatory vertex case 2). This step is essentially the same as Step 5 of the general case. Consider all the r_2'-subwalls of W_1 that do not contain vertices of S', which are at most $\binom{r_1}{r_2'}^2 = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{9/2} \log k)}$ many, and for each of them, say $\tilde{W}_2$, compute its canonical partition $\mathcal{Q}'$. Note that $\tilde{W}_2$ is a wall of $G - S'$, and thus of G'. Then, in G', we contract each bag Q of $\mathcal{Q}'$ to a single vertex v_Q, and add a new vertex v_{all} and make it adjacent to each v_Q such that Q is an internal bag of $\mathcal{Q}'$. In the resulting graph $G'_{\mathcal{Q}'}$, for every vertex v of $G' - V(\tilde{W}_2)$, check, using augmenting paths from usual maximum flow techniques [87], whether there are q internally vertex-disjoint paths from v_{all} to v in time $\mathcal{O}(q \cdot m) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^4\sqrt{\log k} \cdot n)$. If this is the case for some v, then by Lemma 7.3.3, it holds that, for every solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathcal{P}$ for the instance (G, S', H_2', ϕ', k), it holds that $v \in S \setminus S'$ and that $|\phi^+(v)| < |v| = 1$. Hence, we conclude as in **Step 4**.

Note that, if (G, S', H_2', ϕ', k) is a yes-instance, then, by Claim 7.3.7, there is such a v, since otherwise we would have gone to Step 4. Therefore, if there is no such a v, then we report a no-instance.

Running time. Step 1, 2, 3a, 4, 5, 6, and 7 respectively take time $\mathcal{O}_{\ell_{\mathcal{F}}}(1)$, $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^9)} \cdot n$, $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{5/2} \log k)} \cdot n$, $\mathcal{O}_{\ell_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$, $\mathcal{O}_{\ell_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$, $\mathcal{O}_{\ell_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$, and $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{9/2} \log k)} \cdot n^2$. Given that Step 6 can be applied at most k times, since the size of S' increases by one each time, and that the other steps can be applied at most n times, the algorithm thus runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^9)} \cdot n^2$.

7.4 Irrelevant vertex

This section is dedicated to proving Theorem 7.3.1 and Theorem 7.3.2. The irrelevant vertex technique, which originates from [271], essentially consists in finding inside a flat wall W a smaller flat wall W' that is tight and homogeneous (Proposition 4.6.12), and then arguing that the central vertices of W' are irrelevant with respect to the considered problem, in the sense that they can be removed without affecting the type (positive or negative) of the instance (Lemma 7.4.3).

The proof of Lemma 7.4.3 in Subsection 7.4.2 takes inspiration from the proof of [285, Lemma 16], which corresponds to the particular case of VERTEX DELETION TO $\mathsf{exc}(\mathcal{F})$. Indeed, both proofs combine a result of [24] (Proposition 4.6.11) with an auxiliary result (Subsection 7.4.1) claiming the existence, inside a flat wall W with a central vertex v, of a smaller flat wall W' avoiding vertices of a solution aside from its central part, that also contains v. The more general case of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ is however more involved, given that doing some modification is not as straightforward as removing vertices, and that we now have annotations. In particular, it requires to give in Subsection 4.6.4 a new definition of homogeneous flat wall, that encompasses the one used in [24, 235, 284–286].

In the bounded genus case, instead of Proposition 4.6.12, we prove in Lemma 7.4.1 that a flat wall W can be slightly modified to become homogeneous if it respects some additional planar embeddability conditions. Proposition 4.6.12 and Lemma 7.4.1 are the core ingredients explaining the gap in the running time between the general and the bounded genus case. Proposition 4.6.11 requires a flat wall that is both homogeneous and tight. In [24, 235, 284–286], the tightness condition is implicit, given that a flat wall can always is transformed in a tight flat wall (Proposition 4.6.10). In the bounded genus case however, if we transform our homogeneous flat wall, we might lose the homogeneity condition, so we need to explicitly prove that our homogeneous wall is also tight in Lemma 7.4.1. Finally, we prove Theorem 7.3.2 in Subsection 7.4.3.

The size of the flatness pair $(W, \mathfrak{R})$ necessary to find a homogeneous flatness pair in Proposition 4.6.12 is very large and is the main cause of the huge degree of the polynomial in k in the running time of Theorem 7.1.3.

However, in the bounded genus case, we can find a homogeneous wall inside a flatness pair of smaller size if we additionally ask that its compass is embeddable in a disk and that no "leaf-block" of the graph is planar. A *leaf-block* in a graph G is either a connected component C of G, or the graph $G[V(C) \cup \{v\}]$ for some vertex $v \in V(G)$ and some connected component C of $G - v$. Given a leaf-block B, with denote by V_B the set of $V(C)$.

Lemma 7.4.1. *There exists an algorithm with the following specifications:*

`Planar-Homogeneous`$(G, W, \mathcal{R})$
Input*: A graph G whose leaf-blocks are not planar and a flatness pair $(W, \mathfrak{R} = (X, Y, P, C, \rho))$ of G whose $\mathfrak{R}$-compass is embeddable in a disk with $X \cap Y$ on its boundary.*
Output*: A 5-tuple $\mathfrak{R}'$ such that $(W, \mathfrak{R}')$ is a flatness pair of G that is regular, tight, and ℓ-homogeneous with respect to $\emptyset$ for any $\ell \in \mathbb{N}$.*
Moreover, the algorithm runs in time $\mathcal{O}(n + m)$.

Proof. Let Ω be the cyclic ordering of the vertices of $X \cap Y$ as they appear in $D(W)$. Given that $G[Y]$ is embeddable in a disk with $X \cap Y$ on its boundary, there is a vortex-free rendition ρ' of $(G[Y], \Omega)$ in the sphere such that $\pi_{\rho'}(N(\rho')) = V(G)$ and each cell of ρ' contains exactly one edge of G, i.e., for each $c \in C(\rho')$, there is $e \in E(G)$ such that $\sigma_{\rho'}(c) = (e, \{e\})$. Note that, for each $c \in C(\rho')$, $|\tilde{c}| = 2$.

Let us transform ρ' into a tight rendition of $(G[Y], \Omega)$. Note that the only item that is not easily verified is item 5 of the definition of a tight rendition. That is, there might be a $c \in C(\rho')$ such that there are strictly less than $|\tilde{c}| = 2$ vertex-disjoint paths in G from $\pi_{\rho'}(\tilde{c})$ to $V(\Omega)$. Let c be such a cell. If there is no path in G from $\pi_{\rho'}(\tilde{c})$ to $V(\Omega)$, then $\sigma(c)$ belongs to a connected component C of G that does not contain vertices of $V(\Omega)$. Therefore, C is a planar leaf-block of G, a contradiction. Otherwise, there is $v \in V(G) \setminus \pi_{\rho'}(\tilde{c})$ such that every path from $\pi_{\rho'}(\tilde{c})$ to $V(\Omega)$ contains v, so v is a cut vertex of G. Therefore, $\sigma_{\rho'}(c)$ belongs to a connected component C of $G - v$ not containing vertices of $V(\Omega)$, and thus a planar leaf-block of G, again a contradiction. Therefore, ρ' is a tight rendition.

We set $\mathcal{R}' := (X, Y, P, C, \rho')$. Given that $(W, \mathcal{R})$ is a flatness pair of G and that none of the cells of ρ' is W-external, W-marginal, or untidy, we conclude that $(W, \mathcal{R}')$ is also a regular flatness pair of G.

Finally, given that each cell of ρ' contains exactly one edge and has a boundary of size two, we conclude that, for any $\ell \in \mathbb{N}$, the ℓ-folio of $\mathbf{F}$ is the same for each $\mathbf{F} \in \mathsf{Flaps}_{\mathcal{R}}(W)$. Therefore, $(W, \mathcal{R}')$ is ℓ-homogeneous with respect to $\emptyset$, hence the result. $\square$

7.4.1 An auxiliary lemma

The following lemma says that given a big enough flat wall W and a vertex set S of size at most k, there is a smaller flat wall W^* such that the central vertices of W and the intersection of S with the compass of W^* are contained in the compass of the central wall of height five of W^*. This result is used in [285], but is not stated as a stand-alone result, so we reprove it here for completeness.

Lemma 7.4.2. *There exists a function $f_{7.4.2} : \mathbb{N}^3 \to \mathbb{N}$, whose images are odd integers, such that the following holds.*

Let $a, d, k, q, z \in \mathbb{N}$, with odd $q \geq 3$ and odd $z \geq 5$, G be a graph, $S \subseteq V(G)$, where $|S| \leq k$, $(W, \mathfrak{R})$ be a regular and tight flatness pair of G of height at least $f_{7.4.2}(k, z, q)$ that is (a, d)-homogeneous, and $(W', \mathfrak{R}')$ be a $W^{(q)}$-tilt of $(W, \mathfrak{R})$. Then, there is a flatness pair $(W^*, \mathfrak{R}^*)$ of G such that:

- *$(W^*, \mathfrak{R}^*)$ is a $\tilde{W}'$-tilt of $(W, \mathfrak{R})$ for some z-subwall $\tilde{W}'$ of W,*
- *$(W^*, \mathfrak{R}^*)$ is regular, tight, and (a, d)-homogeneous, and*
- *$V(\mathsf{Compass}_{\mathfrak{R}'}(W'))$ and $S \cap V(\mathsf{Compass}_{\mathfrak{R}^*}(W^*))$ are both subsets of the vertex set of the compass of every ${W^*}^{(5)}$-tilt of $(W^*, \mathfrak{R}^*)$.*

Moreover, $f_{7.4.2}(k, z, q) = \mathsf{odd}((k + 1) \cdot (z + 1) + q)$.

Proof. Let $r := f_{7.4.2}(k, z, q)$. For every $i \in [r]$, we denote by P_i (resp. Q_i) the i-th vertical (resp. horizontal) path of W. Let $z' := \frac{z+1}{2}$ and observe that, since z is odd, we have $z' \in \mathbb{N}$. We also define, for every $i \in [k + 1]$ the graph

$$B_i := \bigcup_{j \in [z'-1]} P_{f_{z'}(i,j)} \cup \bigcup_{j \in [z']} P_{r+1-f_{z'}(i,j)} \cup \bigcup_{j \in [z'-1]} Q_{f_{z'}(i,j)} \cup \bigcup_{j \in [z']} Q_{r+1-f_{z'}(i,j)},$$

where $f_{z'}(i, j) := j + (i - 1) \cdot (z' + 1)$. For every $i \in [k + 1]$, we define W_i to be the graph obtained from B_i after repeatedly removing from B_i all vertices of degree one (see Figure 7.7 for an example). Since $z = 2z' - 1$, for every $i \in [k + 1]$, W_i is a z-subwall of W. For every $i \in [k + 1]$, we set L^i_{in}

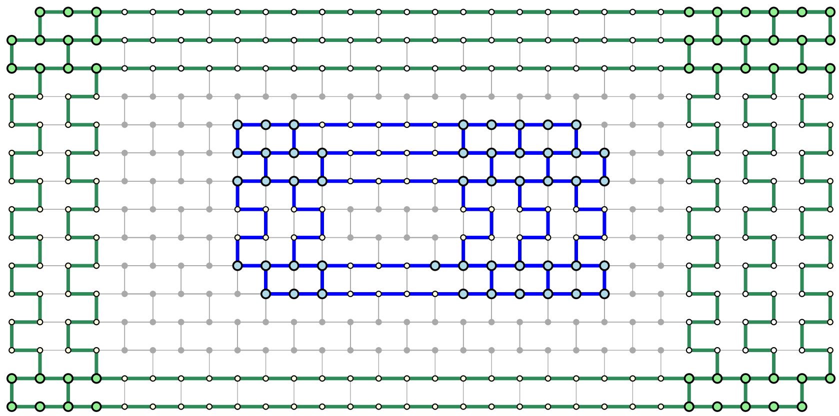

Figure 7.7: A 15-wall and the 5-walls W_1 and W_2 as in the proof of Lemma 7.4.3, depicted in green and blue, respectively. The white vertices are subdivision vertices of the walls W_1 and W_2. This figure is adapted from [285, Figure 3].

to be the inner layer of W_i. Notice that L^i_{in}, for $i \in [k+1]$, and $D(W^{(q)})$ are $\mathfrak{R}$-normal cycles of $\mathsf{Compass}_{\mathfrak{R}}(W)$.

By definition of a tilt of a flatness pair, it holds that $V(\mathsf{Compass}_{\mathfrak{R}'}(W')) \subseteq V(\bigcup \mathsf{Influence}_{\mathfrak{R}}(W^{(q)}))$. Moreover, for every $i \in [k+1]$, the fact that $r \geq (k+1)\cdot(z+2)+q$ implies that $\bigcup \mathsf{Influence}_{\mathfrak{R}}(W^{(q)})$ is a subgraph of $\bigcup \mathsf{Influence}_{\mathfrak{R}}(L^i_{\mathsf{in}})$. Hence, for every $i \in [k+1]$, we have that $V(\mathsf{Compass}_{\mathfrak{R}'}(W')) \subseteq V(\bigcup \mathsf{Influence}_{\mathfrak{R}}(L^i_{\mathsf{in}}))$.

For every $i \in [k+1]$, let $(W'_i, \mathfrak{R}_i)$ be a flatness pair of G that is a W_i-tilt of $(W, \mathfrak{R})$ (which exists due to Proposition 4.6.6). Also, note that, for every $i \in [k+1]$, L^i_{in} is the inner layer of W_i and therefore it is an $\mathfrak{R}_i$-normal cycle of $\mathsf{Compass}_{\mathfrak{R}_i}(W'_i)$. Additionally, for every $i \in [k+1]$, $(W'_i, \mathfrak{R}_i)$ is (a,d)-homogeneous due to Observation 4.6.9, and, due to Observation 4.6.5, $(W_i, \mathfrak{R}_i)$ is also regular. Also, by Proposition 4.6.10, we can assume $(W'_i, \mathfrak{R}_i)$ to be tight.

For every $i \in [k+1]$, we set $D_i := V(\mathsf{Compass}_{\mathfrak{R}_i}(W'_i)) \setminus V(\bigcup \mathsf{Influence}_{\mathfrak{R}_i}(L^i_{\mathsf{in}}))$. Given that the vertices of $V(W_i)$ are contained between the $((i-1)\cdot(z'+1)+1)$-th and the $(i\cdot(z'+1)-1)$-th layers of W for $i \in [k+1]$, it implies that the vertex sets D_i, $i \in [k+1]$, are pairwise disjoint. Therefore, since that $|S| \leq k$, there exists a $j \in [k+1]$ such that $S \cap D_j = \emptyset$. Thus, $S \cap V(\mathsf{Compass}_{\mathfrak{R}_j}(W'_j)) \subseteq V(\bigcup \mathsf{Influence}_{\mathfrak{R}_j}(L^j_{\mathsf{in}}))$.

Let Y be the vertex set of the compass of some $W_j^{(5)}$-tilt of $(W'_j, \mathfrak{R}_j)$. Note that L^j_{in} is the perimeter of $W_j^{(3)}$, and therefore, we have $S \cap V(\mathsf{Compass}_{\mathfrak{R}_j}(W'_j)) \subseteq V(\bigcup \mathsf{Influence}_{\mathfrak{R}_j}(L^j_{\mathsf{in}})) \subseteq Y$ and $V(\mathsf{Compass}_{\mathfrak{R}'}(W')) \subseteq V(\bigcup \mathsf{Influence}_{\mathfrak{R}}(L^j_{\mathsf{in}})) \subseteq Y$. Therefore, $(W'_j, \mathfrak{R}_j)$ is the desired flatness pair. $\square$

7.4.2 Finding an irrelevant vertex in a homogeneous flat wall

The next lemma states that the central vertex of a big enough homogeneous flat wall is irrelevant. As mentioned previously, Lemma 7.4.3 takes inspiration from [285, Lemma 16], though the proof is more involved in Lemma 7.4.3, due to the more general modifications allowed and the annotation.

Lemma 7.4.3. *Let $\mathcal{F}$ be a finite collection of graphs and $\mathcal{L}$ be a hereditary R-action. There exists a function $f_{7.4.3} : \mathbb{N}^4 \to \mathbb{N}$, whose images are odd integers, such that the following holds.*

Let $k, q, a \in \mathbb{N}$, with odd $q \geq 3$. Let G be a graph, $S' \subseteq V(G)$ be a set of size at most k, $(H'_2, \phi') \in \mathcal{L}(G[S'])$, and $G' := G^{S'}_{(H'_2,\phi')}$. Let $A \subseteq V(G')$ be a subset of size at most a, $(W, \mathfrak{R})$ be a *regular and tight flatness pair of $G' - A$ of height at least $f_{7.4.3}(a, \ell_{\mathcal{F}}, q, k)$ that is $(a, g_{4.6.11}(a, \ell_{\mathcal{F}}))$-homogeneous and such that $\phi'(S') \cap V(\mathsf{Compass}_{\mathfrak{R}}(W)) = \emptyset$. Let $(W', \mathfrak{R}')$ be a $W^{(q)}$-tilt of $(W, \mathfrak{R})$ and $Y := V(\mathsf{Compass}_{\mathfrak{R}'}(W'))$.*

Then, (G, S', H_2', ϕ', k) and $(G - Y, S', H_2', \phi', k)$ are equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$.

Moreover, $f_{7.4.3}(a, \ell_{\mathcal{F}}, q, k) = f_{7.4.2}(k, f_{4.6.11}(a, \ell_{\mathcal{F}}, 5), q)$.

Proof. Let $z := f_{4.6.11}(a, \ell_{\mathcal{F}}, 5)$, $d := g_{4.6.11}(a, \ell_{\mathcal{F}})$, and $r =: f_{7.4.3}(a, \ell_{\mathcal{F}}, q, k) = f_{7.4.2}(k, z, q)$.

The forward direction is immediate given that $\mathcal{L}$ is hereditary. Indeed, suppose (S, H_2, ϕ) is a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k). Let $S^* = S \setminus Y \subseteq V(G) \setminus Y$. Then, because $\mathcal{L}$ is hereditary, the restriction of (H_2, ϕ) to S^* is in $\mathcal{L}((G - Y)[S^*])$. Given that $Y \subseteq V(\mathsf{Compass}_{\mathfrak{R}}(W)) \subseteq V(G') \setminus \phi^+(S') = V(G) \setminus S'$, it follows that $S' \cap Y = \emptyset$, and thus that $S' \subseteq S^*$. Therefore, the restriction of (H_2^*, ϕ^*) to S' is (H_2', ϕ'). Moreover, $G^* := (G - Y)^{S^*}_{(H_2^*, \phi^*)} = G^S_{(H_2, \phi)} - Y$, so $G^* \in \mathsf{exc}(\mathcal{F})$. We conclude that (S^*, H_2^*, ϕ^*) is a solution of $(G - Y, S', H_2', \phi', k)$.

Suppose now that (S, H_2, ϕ) is a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance $(G - Y, S', H_2', \phi', k)$. Let us now prove that there is $S^* \subseteq S$ such that $(S^*, H_2[\phi^+(S^*)], \phi|_{S^*})$ is a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k).

By Lemma 7.4.2, there exists a regular and tight flatness pair $(W^*, \mathfrak{R}^*)$ of $G' - A$ of height z that is d-homogeneous with respect to 2^A such that Y and $S \cap Y^*$ are both subsets of the vertex set of the compass of every $W^{*(5)}$-tilt of $(W^*, \mathfrak{R}^*)$, where $Y^* = V(\mathsf{Compass}_{\mathfrak{R}^*}(W^*))$. Additionally, $Y \subseteq V(\mathsf{Compass}_{\mathfrak{R}}(W))$, so $S' \cap Y^* = \emptyset$. Let $S^* = S \setminus Y^* \supseteq S'$. Given that $\mathcal{L}$ is hereditary, the restriction (H_2^*, ϕ^*) of (H_2, ϕ) to S^* is in $\mathcal{L}(G[S^*])$. Moreover, the restriction of (H_2^*, ϕ^*) to S' is (H_2', ϕ'). It thus remains to prove that $G^* := G^{S^*}_{(H_2^*, \phi^*)} \in \mathsf{exc}(\mathcal{F})$.

Let Y' be the compass of some $W^{*(5)}$-tilt of $(W^*, \mathfrak{R}^*)$. Hence we have $Y, S \cap Y^* \subseteq Y'$ and $S^* \cap Y' \neq \emptyset$. Therefore, we have $G^* - Y' = (G - Y)^S_{(H_2, \phi)} - Y'$. Given that $(G - Y)^S_{(H_2, \phi)} \in \mathsf{exc}(\mathcal{F})$ and that $\mathsf{exc}(\mathcal{F})$ is minor-closed, it implies that $G^* - Y' \in \mathsf{exc}(\mathcal{F})$.

Claim 7.4.4. *Y' is $\ell_{\mathcal{F}}$-irrelevant in G^*.*

Proof of claim. Let $S_r = S^* \setminus S'$. We set $A^* := A \setminus S_r \cup \phi^+(A \cap S_r)$. If $\mathfrak{R}^* = (X^*, Y^*, P, C, \Gamma, \sigma, \pi)$, then we set $\mathfrak{R}^*_{\mathcal{L}} = (X^*_{\mathcal{L}}, Y^*, P, C, \Gamma, \sigma, \pi)$, where $X^*_{\mathcal{L}} = (X^* \setminus S_r) \cup \phi^+(S_r) \setminus A^*$. Given that $S_r \cap Y^* = \emptyset$, it implies that $(W^*, \mathfrak{R}^*_{\mathcal{L}})$ is a flatness pair of $G^* - A^*$. Notice that the $\mathfrak{R}^*$-compass and the $\mathfrak{R}^*_{\mathcal{L}}$-compass of W^* are identical, which implies that $(W^*, \mathfrak{R}^*_{\mathcal{L}})$ is a regular flatness pair of $G^* - A^*$ that is (a, d)-homogeneous. Given that $|A^*| \leq |A| \leq a$, it implies in particular that $(W^*, \mathfrak{R}^*_{\mathcal{L}})$ is a regular flatness pair of $G^* - A^*$ that is d-homogeneous with respect to A^*.

Given that $z = f_{4.6.11}(a, \ell_{\mathcal{F}}, 5)$ and $d := g_{4.6.11}(a, \ell_{\mathcal{F}})$, we can thus apply Proposition 4.6.11 with input $(a, \ell_{\mathcal{F}}, 5, G^*, A^*, (W^*, \mathfrak{R}^*_{\mathcal{L}}))$ which implies that the vertex set of the compass of every $W^{*(5)}$-tilt of $(W^*, \mathfrak{R}^*_{\mathcal{L}})$ is $\ell_{\mathcal{F}}$-irrelevant. This is in particular the case of Y', given that the $\mathfrak{R}^*$-compass and the $\mathfrak{R}^*_{\mathcal{L}}$-compass of W^* are identical. $\diamond$

Given that Y' is $\ell_{\mathcal{F}}$-irrelevant in G^* by Claim 7.4.4 and that every graph in $\mathcal{F}$ has detail at most $\ell_{\mathcal{F}}$, it implies that $G^* \in \mathsf{exc}(\mathcal{F})$ if and only if $G^* - Y' \in \mathsf{exc}(\mathcal{F})$, hence the result. $\square$

After combining Proposition 4.6.12 and Lemma 7.4.3, we finally get our algorithm to find an irrelevant vertex inside a flat wall in the general case.

Proof of Theorem 7.3.1. Let $r = f_{7.4.3}(a, \ell_{\mathcal{F}}, 3, k)$, $\ell = g_{4.6.11}(a, \ell_{\mathcal{F}})$, and $f_{7.3.1}(k, a) = f_{4.6.12}(r, a, \ell) = \mathcal{O}_{a, \ell_{\mathcal{F}}}(k^c)$.

We apply the algorithm `Homogeneous` of Proposition 4.6.12 with input $(r, a, \ell, t, G' - A, W, \mathcal{R})$. It outputs in time $2^{\mathcal{O}(g_{4.6.12}(a,\ell)\cdot r\log r + t\log t)} \cdot (n+m)$ a flatness pair $(\breve{W}, \breve{\mathfrak{R}})$ of $G' - A$ of height r that is tight, (a,ℓ)-homogeneous, and is a W'-tilt of $(W, \mathfrak{R})$ for some subwall W' of W. Given that $(W, \mathfrak{R})$ is a regular flatness pair, by Observation 4.6.5, so is $(\breve{W}, \breve{\mathfrak{R}})$. Given that $r = f_{7.4.3}(a, \ell_{\mathcal{F}}, 3, k)$, that $\ell = g_{4.6.11}(a, \ell_{\mathcal{F}})$, and that $\phi'(S')$ does not intersect the $\mathfrak{R}$-compass of W, and thus neither the $\breve{\mathfrak{R}}$-compass of $\breve{W}$, we conclude by Lemma 7.4.3, that for any $W^{(3)}$-tilt $(W', \mathfrak{R}')$ of $(\breve{W}, \breve{\mathfrak{R}})$, (G, S', H_2', ϕ', k) and $(G - Y, S', H_2', \phi', k)$ are equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$, where $Y := V(\mathsf{Compass}_{\mathfrak{R}'}(W'))$. Let v be a central vertex of $\breve{W}$. Given that $v \in Y$, (G, S', H_2', ϕ', k) and $(G - v, S', H_2', \phi', k)$ are in particular equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$. Hence the result. $\square$

7.4.3 Irrelevant vertex in the bounded genus case

The next lemma essentially states that planar leaf-blocks are irrelevant in the bounded genus case.

Lemma 7.4.5. *Let $\mathcal{L}$ be a hereditary R-action and $\mathcal{F}$ be the collection of obstructions of the graphs embeddable in a surface Σ of genus at most g. Let G be a graph and $k \in \mathbb{N}$. Let $S' \subseteq V(G)$ be a set of size at most k and $(H_2', \phi') \in \mathcal{L}(G[S'])$. Suppose that there is a planar leaf-block B of $G' := G^{S'}_{(H_2',\phi')}$ such that $V_B \cap \phi'(S') = \emptyset$. Then, (G, S', H_2', ϕ', k) and $(G - V_B, S', H_2', \phi', k)$ are equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$.*

Proof. Suppose that there is a solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k). Let $S^* := S \setminus V_B$. Note that $S' \subseteq S^*$. Let (H_2^*, ϕ^*) be the restriction of (H_2, ϕ) to S^*. Given that $\mathcal{L}$ is hereditary, $(H_2^*, \phi^*) \in \mathcal{L}(G[S^*])$. Given that $(G - V_B)^{S^*}_{(H_2^*,\phi^*)} = G^S_{(H_2,\phi)} - V_B$ is a subgraph of $G^S_{(H_2,\phi)} \in \mathsf{exc}(\mathcal{F})$ and that $\mathsf{exc}(\mathcal{F})$ is hereditary, it implies that $(G - V_B)^{S^*}_{(H_2^*,\phi^*)} \in \mathsf{exc}(\mathcal{F})$. So (S^*, H_2^*, ϕ^*) is a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance $(G - V_B, S', H_2', \phi', k)$.

Suppose now that there is a solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance $(G - V_B, S', H_2', \phi', k)$. Let $G'' := (G - V_B)^S_{(H_2,\phi)}$ and $G^* := G^S_{(H_2,\phi)}$. Note that G^* is obtained by taking the disjoint union of G'' and B and identifying at most one vertex of both sides (that is, the vertex $v \in V(B) \setminus V_B$, if it exists). Given that G'' is embeddable in Σ and that B is planar, we conclude that G^* is embeddable in Σ as well. Therefore, (S, H_2, ϕ) is a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k). $\square$

After combining Lemma 7.4.1, Lemma 7.4.3, and Lemma 7.4.5, we finally get our algorithm to find an irrelevant vertex inside a flat wall in the bounded genus case.

Proof of Theorem 7.3.2. Let $r = f_{7.3.2}(k) := f_{7.4.3}(0, \ell_{\mathcal{F}}, 3, k)$ and $\ell = g_{4.6.11}(a, \ell_{\mathcal{F}})$.

We can find all the cut vertices of G' using a depth-first search algorithm in time $\mathcal{O}(n+m)$. Therefore, if there is a planar leaf-block B in G', then we can find it in time $\mathcal{O}(n+m)$. In that case, we can return V_B by Lemma 7.4.5.

Otherwise, we apply the algorithm `Planar-Homogeneous` of Lemma 7.4.1 with input $(G', W, \mathfrak{R})$, which outputs a 5-tuple $\mathfrak{R}'$ such that $(W, \mathfrak{R}')$ is a flatness pair of G' that is regular, tight, and $(0,\ell)$-homogeneous. Given that $r = f_{7.4.3}(0, \ell_{\mathcal{F}}, 3, k)$, that $\ell = g_{4.6.11}(a, \ell_{\mathcal{F}})$, and that $\phi'(S')$ does not intersect the $\mathfrak{R}$-compass of W, which is also the $\mathfrak{R}'$-compass of W, we conclude by Lemma 7.4.3 that for any $W^{(3)}$-tilt $(W^*, \mathfrak{R}^*)$ of $(W, \mathfrak{R}')$, (G, S', H_2', ϕ', k) and $(G - Y, S', H_2', \phi', k)$ are equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$, where $Y := V(\mathsf{Compass}_{\mathfrak{R}^*}(W^*))$. Let v be a central vertex of W^*. Given that $v \in Y$, (G, S', H_2', ϕ', k) and $(G - v, S', H_2', \phi', k)$ are in particular equivalent instances of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$. So we can return $Y := \{v\}$, hence the result. $\square$

7.5 Obligatory sets

This section is dedicated to proving Lemma 7.3.3. The proof is quite similar to [285, Lemma 13], with a more involved notation. However we require, and thus prove a stronger result, which is that, not only we find a set A containing a vertex in the solution, but the size of this set A must decrease after the modification. We prove in Lemma 7.5.3 that if G contains a complete A-apex grid as an A-fixed minor (see below for the definitions), then A intersects any solution S. More specifically, after the modification of G restricted to A is done, the size of G must decrease. We then derive Lemma 7.3.3 from Lemma 7.5.3, that is merely a translation that helps us in our setting of walls to easily find such an A-fixed minor.

Apex grids. Let H be a graph, $A \subseteq V(H)$, and $r \in \mathbb{N}$. H is an *A-apex r-grid* if $H - A$ is a r-grid. H is a *complete A-apex r-grid* if it is a A-apex r-grid and that there is an edge between each vertex of A and each vertex of $H - A$.

Fixed minors. Given a graph G and a set $A \subseteq V(G)$, we say that a graph H is a *A-fixed minor* of G if H can be obtained from a subgraph G' of G where $A \subseteq V(G')$ after contracting edges without endpoints in A. For example, the graph of Figure 7.3 contains an A-apex 3-grid as an A-fixed minor.

The following result says that a complete A-apex grid is always an A-fixed minor of a big enough A-apex grid Γ such that each vertex of A has sufficiently many neighbors in the central part of Γ.

Proposition 7.5.1 ([285]). *There exist three functions $f_{7.5.1}, g_{7.5.1} : \mathbb{N}^2 \to \mathbb{N}$, and $h_{7.5.1} : \mathbb{N} \to \mathbb{N}$ such that if $r, a \in \mathbb{N}$, H is an A-apex h-grid, where $A \subseteq V(H)$ has size at most a, $h \geq f_{7.5.1}(r,a) + 2 \cdot h_{7.5.1}(r)$, and each vertex of A has at least $g_{7.5.1}(r,a)$ neighbors in the central $f_{7.5.1}(r,a)$-grid of $H - A$, then H contains as an A-fixed minor a complete A-apex r-grid.*

Moreover, $f_{7.5.1}(r,a) = \mathcal{O}(r^4 \cdot 2^a)$, $g_{7.5.1}(r,a) = \mathcal{O}(r^6 \cdot 2^a)$, and $h_{7.5.1}(r) = \mathcal{O}(r^2)$.

The following easy observation intuitively states that every planar graph H is a minor of a big enough grid, where the relationship between the size of the grid and $|V(H)|$ is linear (see e.g., [282]).

Proposition 7.5.2. *There exists a function $f_{7.5.2} : \mathbb{N} \to \mathbb{N}$ such that every planar graph on n vertices is a minor of the $f_{7.5.2}(n)$-grid. Moreover, $f_{7.5.2}(n) = \mathcal{O}(n)$.*

We now prove that, if G contains a complete A-apex grid as an A-fixed minor, then A intersects any solution S, and that the partial modification of A decreases the size of the graph.

Lemma 7.5.3. *There exists a function $f_{7.5.3} : \mathbb{N} \to \mathbb{N}$ such that the following holds.*

Let $\mathcal{F}$ be a finite collection of graphs, $\mathcal{L}$ be a hereditary R-action, and $k \in \mathbb{N}$. Let G be a graph, $S' \subseteq V(G)$ be a set of size at most k, and $(H_2', \phi') \in \mathcal{L}(G[S'])$. Suppose that $G' := G^{S'}_{(H_2', \phi')}$ contains a complete A-apex $f_{7.5.3}(k)$-grid H as an A-fixed minor for some $A \subseteq V(G')$ with $|A| = a_{\mathcal{F}}$.

Then, for every solution (S, H_2, ϕ) of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for (G, S', H_2', ϕ'), it holds that $A' \neq \emptyset$, where $A' := (S \setminus S') \cap A$, and that $|\phi^+(A')| < |A'|$.

Moreover $f_{7.5.3}(k) = \mathcal{O}(\sqrt{(k + {a_{\mathcal{F}}}^2 + 1)} \cdot s_{\mathcal{F}})$.

Proof. Let $d = f_{7.5.2}(s_{\mathcal{F}} - a_{\mathcal{F}})$ and $r = \lceil \sqrt{(k + a_{\mathcal{F}}^2 + 1)} \cdot d \rceil$. We set $f_{7.5.3}(a_{\mathcal{F}}, s_{\mathcal{F}}, k) = r$ and we notice that, since $d = \mathcal{O}(s_{\mathcal{F}})$, it holds that $f_{7.5.3}(k) = \mathcal{O}(\sqrt{(k + a_{\mathcal{F}}^2 + 1)} \cdot s_{\mathcal{F}})$.

Observe that since $r = \lceil\sqrt{(k + a_{\mathcal{F}}^2 + 1)} \cdot d\rceil$, $V(H \setminus A)$ can be partitioned into $(k + a_{\mathcal{F}}^2 + 1)$ vertex sets $V_1, \ldots, V_{k+a_{\mathcal{F}}^2+1}$ such that, for every $i \in [k + a_{\mathcal{F}}^2 + 1]$, the graph $H[V_i]$ is a d-grid.

Let $\{S_v \mid v \in V(H)\}$ be a model of H in G'. Let $S^* := S \setminus S' \cup \phi^+(S')$. There exists a pair $(H_2^*, \phi^*) \in \mathcal{M}(G'[S^*])$ such that $G^* := G^S_{(H_2,\phi)} = G'^{S^*}_{(H_2^*,\phi^*)}$. Note in particular that, given that (H_2', ϕ') is the restriction of (H_2, ϕ) to S', it implies that $\phi^*|_{\phi^+(S')} = \mathsf{id}_{\phi^+(S')}$ and that $\phi^*|_{V(G')\setminus\phi^+(S')} = \phi|_{V(G')\setminus\phi^+(S')}$. Given that $|S^*| \leq |S| \leq k$, there is $I \subseteq [k + a_{\mathcal{F}}^2 + 1]$ of size at least $a_{\mathcal{F}}^2 + 1$ such that for each $i \in I$, for each $v \in V_i$, $S_v \cap S^* = \emptyset$. Hence, $H' := H[A \cup \bigcup_{i\in I} V_i]$ is a minor of G' such that $V(H') \cap S^* \subseteq A$.

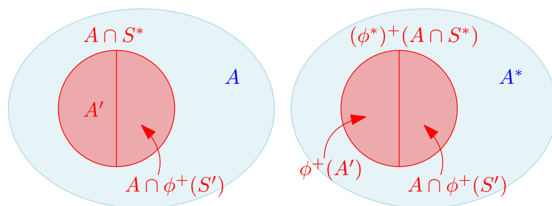

Figure 7.8: The sets A and A^* in the proof of Lemma 7.5.3.

Suppose towards a contradiction that $|\phi^+(A')| = |A'|$. Let $A^* := (A \setminus S^*) \cup (\phi^*)^+(S^*)$ be the set of G^* obtained from A after the modification (H_2^*, ϕ^*). Given that $|A^*| - |A| = |\phi^+(A')| - |A'|$ (see Figure 7.8 for an illustration), this implies that $|A^*| = |A|$. Intuitively, this means that the modification (H_2^*, ϕ^*) only deleted or added edges of $G'[A]$, but no vertices where deleted and no two vertices were identified together. Therefore, the graph H^* obtained from H' by removing the edges between the vertices of A is a minor of G^*.

Let F be a graph in $\mathcal{F}$ of apex number $a_{\mathcal{F}}$. We fix $i \in I$. Given that $d = f_{7.5.2}(s_{\mathcal{F}} - a_{\mathcal{F}})$, Proposition 7.5.2 implies that every planar graph on $s_{\mathcal{F}} - a_{\mathcal{F}}$ vertices is a minor of $H[V_i]$. Additionally, given that $|I \setminus \{i\}| = a_{\mathcal{F}}^2$ and that, for each $j \in I \setminus \{i\}$, $H''[V_j \cup A]$ is a complete A-apex d-grid, for each pair of vertices in A, we can find a path connecting them through some $H''[V_j \cup A]$, and thus it implies that every graph on $a_{\mathcal{F}}$ vertices is a minor of $H'' - V_i$. Therefore, F is a minor of H'', and hence of G^*, a contradiction. □

We finally prove the main result of the section.

Proof of Lemma 7.3.3. Let $r := f_{7.5.3}(k)$, $h = f_{7.3.3}(k) := f_{7.5.1}(r, a_{\mathcal{F}}) + 2 \cdot h_{7.5.1}(r) + 2$, $q = g_{7.3.3}(k) := g_{7.5.1}(r, a_{\mathcal{F}})$, and $p = h_{7.3.3}(k) := f_{7.5.1}(r, a_{\mathcal{F}})$. Note that $r = \mathcal{O}_{s_{\mathcal{F}}}(k^{1/2})$, and thus that $h = \mathcal{O}_{s_{\mathcal{F}}}(k^2)$, $q = \mathcal{O}_{s_{\mathcal{F}}}(k^3)$, and $p = \mathcal{O}_{s_{\mathcal{F}}}(k^2)$. Let $G'_{\tilde{\mathcal{Q}}}$ be the graph obtained from G' by contracting each bag Q of $\tilde{\mathcal{Q}}$ to a single vertex v_Q. Then $G'_{\tilde{\mathcal{Q}}}$ contains an A-apex $(h-2)$-grid H as a subgraph such that each vertex of A has at least $g_{7.5.1}(r, a_{\mathcal{F}})$ neighbors in the central $f_{7.5.1}(r, a_{\mathcal{F}})$-grid of $H - A$. Therefore, G' contains H as an A-fixed minor. By Proposition 7.5.1, given that $h - 2 = f_{7.5.1}(r, a_{\mathcal{F}}) + 2 \cdot h_{7.5.1}(r)$, $q = g_{7.5.1}(r, a_{\mathcal{F}})$, and $p = f_{7.5.1}(r, a_{\mathcal{F}})$, it implies G contains a complete A-apex r-grid as an A-fixed minor. Therefore, we can conclude using Lemma 7.5.3, given that $r = f_{7.5.3}(k)$. □

7.6 The case of bounded treewidth

In this section, we present a dynamic programming algorithm in the case where the input graph has bounded treewidth.

Theorem 7.3.4. *Let $\mathcal{F}$ be a finite collection of graphs and $\mathcal{L}$ be an R-action. There is an algorithm that, given $k \in \mathbb{N}$, a graph G of treewidth at most w, a set $S' \subseteq V(G)$ of size at most k, and $(H_2', \phi') \in \mathcal{L}(G[S'])$, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2+(k+w)\log(k+w))} \cdot n$ either outputs a solution of $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$ for the instance (G, S', H_2', ϕ', k), or reports a* no*-instance.*

Our dynamic programming algorithm essentially goes as follows. Let G be the input graph of treewidth w. By Proposition 4.3.1 and Proposition 4.3.3, we compute a nice tree decomposition $\mathcal{T} = (T, \beta, r)$ of G of width at most $2w+1$ with $\mathcal{O}(w \cdot n)$ nodes in time $2^{\mathcal{O}(k)} \cdot w^2 \cdot n$. Let (S, H_2, ϕ) be a solution of $\mathcal{L}$-R-$\mathsf{exc}(\mathcal{F})$ for (G, k). Then, for each node of T, in a leaf-to-root manner, we guess the restriction (H_2', ϕ') of (H_2, ϕ) to the graph G_t induced by the subtree of T rooted at t. That is, each time we introduce a vertex v, we guess whether v belongs to S or not, and if we guess that it does, we also guess how it is modified: it can either be deleted ($\phi'(v) = \emptyset$), or identified to a vertex ($\phi'(v) = \phi'(u)$ for some $u \in V(G_t) \setminus \{v\}$), or it can be a new vertex in H_2' (when $\phi'^{-1}(\phi'(v)) = \{v\}$). In this latter case, we also need to guess the edges between $\phi'(v)$ and $u \in V(H_2') \setminus \{\phi'(v)\}$ to get H_2'. Obviously, each time, we need to check that the guessed partial solution (S', H_2', ϕ') is such that $|S'| \leq k$ and that $(G_t)^{S'}_{(H_2',\phi')} \in \mathsf{exc}(\mathcal{F})$, otherwise we reject this guess. After the dynamic programming, we add a post-processing step to keep only the set $\mathcal{A}$ of guessed solutions (S, H_2, ϕ) for the root such that $(H_2, \phi) \in \mathcal{L}(G[S])$. Hence, we have a yes-instance if and only if $\mathcal{A} \neq \emptyset$. For ease of notation, we only formally write our dynamic programming algorithm for $\mathcal{L}$-R-$\mathsf{exc}(\mathcal{F})$. It can be easily adapted to the annotated version $\mathcal{L}$-AR-$\mathsf{exc}(\mathcal{F})$, by simply rejecting tuples that do not follow the annotation during the introduce and the join operations.

The number of possible partial solutions generated by the above description is too big to store them all, given that there is $\binom{n}{k}$ choices for the set S of vertices involved in the modification. Therefore, we instead store a "signature" of G_t that keeps only the necessary information to ensure that each partial solution is represented by an element of the signature, that each element of the signature represents at least one partial solution, and that the signature at a node $t \in V(T)$ can be deduced from those of its children. In order to bound the number of elements in a signature while still being able to check that a guessed partial solution (S', H_2', ϕ') at node t is such that $(G_t)^{S'}_{(H_2',\phi')} \in \mathsf{exc}(\mathcal{F})$, we use the representative-based technique of [24]. Let $\mathbf{G}_t$ be a boundaried graph with underlying graph G_t and the bag $\beta(t)$ as its boundary. This technique essentially guarantees that, if $G_t \in \mathsf{exc}(\mathcal{F})$, then we can replace $\mathbf{G}_t$ with a boundaried graph $\mathbf{R}$ with the same boundary but of smaller size, called representative of $\mathbf{G}_t$, such that, for any boundaried graph $\mathbf{H}$, $\mathbf{H} \oplus \mathbf{G_t} \in \mathsf{exc}(\mathcal{F})$ if and only if $\mathbf{H} \oplus \mathbf{R} \in \mathsf{exc}(\mathcal{F})$ (see below for the definition of $\oplus$). In our case, G_t does not necessarily belong to $\mathsf{exc}(\mathcal{F})$, but we know that $(G_t)^{S'}_{(H_2',\phi')}$ does. Therefore, we remember a representative $\mathbf{R}$ of $G' := (G_t)^{S'}_{(H_2',\phi')}$ in the signature. As said above, when introducing a vertex v, we may guess that $v \in S'$ and that $\phi'(v) = \phi'(u)$ for some $u \in V(G_t) \setminus \{v\}$. Therefore, we need to remember each vertex of G' that is a modified vertex (that is in $(\phi')^+(S')$). Thus, the boundaried graph that we consider is a boundaried graph $\mathbf{G}'$ with underlying graph G' and boundary $\beta(t) \cup (\phi')^+(S')$. Moreover, to remember which vertex of the boundary is in $(\phi')^+(S')$, we reserve them the labels in $[k]$. So H_2 is the graph induced by the vertices with such labels in R. Additionally, to be able to check that $|S'| \leq k$ and that $(H_2', \phi') \in \mathcal{L}(G[S'])$, we remember the graph $H_1' := G[S']$. Finally, to be able to construct the extension of H_1' when adding a new vertex v in S, we remember the vertices of S' that may be adjacent to v, that is $S_B := S \cap \beta(t)$. Therefore, the signature of $\mathbf{G}_t$ is the set of all such $(\mathbf{R}, H_1', \phi', S_B)$. See Subsection 7.6.1 for a formal definition of the signature, and Figure 7.9 for an illustration. The way to construct a signature from its children for leaf (in this case, without children), forget, introduce, and join nodes is explained in Subsection 7.6.2.

7.6.1 Signature

Let $\mathcal{F}$ be a finite collection of graphs, $\mathcal{L}$ be a R-action, and $k, w \in \mathbb{N}$. Let $\mathcal{R}$ be the set of representatives in $\mathcal{R}^{k+w}_{\ell_{\mathcal{F}}}$ whose underlying graphs are $\mathcal{F}$-minor-free.

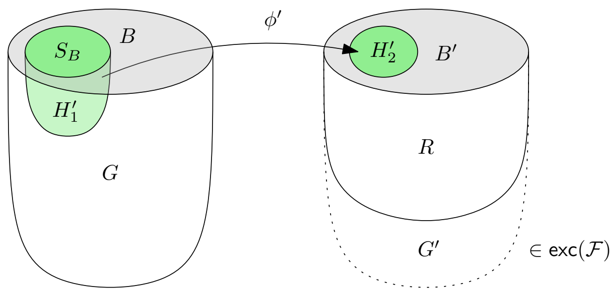

Figure 7.9: An element $(\mathbf{R} = (R, B', \rho'), H_1, \phi, S_B)$ in the signature of $\mathbf{G} = (G, B, \rho)$.

Let $\mathbf{G} = (G, B, \rho)$ be a w-boundaried graph with no label in $[k]$. We call *signature* of $\mathbf{G}$ the set of all tuples $(\mathbf{R} = (R, B', \rho'), H'_1, \phi', S_B)$ such that there exists a set $S' \subseteq V(G)$ of size at most k, there exists a graph H'_2 such that $(H'_2, \phi') \in \mathcal{M}(H'_1)$ and $G' := G^{S'}_{(H'_2,\phi')} \in \mathsf{exc}(\mathcal{F})$, and there exists an injective function $\varphi : (\phi')^+(S') \mapsto [k]$ that is such that:

- $G[S'] = H'_1$ and $R[(\phi')^+(S')] = H'_2$,
- $S_B = S' \cap B$,
- $B' = (B \setminus S_B) \cup (\phi')^+(S')$,
- ρ' is the function such that $\rho'|_{B \setminus S_B} = \rho|_{B \setminus S_B}$ and $\rho'|_{(\phi')^+(S')} = \varphi$, and
- $\mathbf{R}$ is the representative in $\mathcal{R}$ of (G', B', ρ').

See Figure 7.9 for an illustration.

Let us give an upper bound on the number of tuples $(\mathbf{R}, H'_1, \phi', S_B)$ in the signature of $\mathbf{G}$. By Proposition 4.4.2, there are $2^{\mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$ choices for $\mathbf{R}$. Given that H'_1 has at most k vertices, there are at most $2^{\binom{k}{2}}$ choices for H'_1 and at most k^k choices for $\phi : V(H'_1) \to V(H'_2) \cup \{\emptyset\}$ (if $|V(H'_1)| = |V(H'_2)|$, then $\phi : V(H'_1) \to V(H_2)'$ must be a bijection, and otherwise $|V(H'_1)| \geq |V(H'_2) \cup \{\emptyset\}|$). Finally, there are $\binom{w}{\leq k}$ choices for S_B. Hence, the number of tuples is at most $2^{\binom{k}{2} + \mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$.

7.6.2 Dynamic programming

Let G be a graph and let $\mathcal{T} = (T, \beta, r)$ be a nice tree decomposition of G of width w. Let $\rho_0 : V(G) \to [|k+1, k+V(G)|]$ be a bijection. For $t \in V(T)$, we define by G_t the graph induced by the subtree of T rooted at t and by $\mathbf{G_t}$ the boundaried graph $(G_t, \beta(t), \rho_t)$, where $\rho_t := \rho_0|_{\beta(t)}$.

Note that, for each element $(\mathbf{R} = (R, B, \rho), H_1, \phi, S_B)$ of the signature of $\mathbf{G}_r$, given that $G_r = G$, there is $S \subseteq V(G)$ of size at most k such that $H_1 = G[S]$ and $G^S_{(H_2,\phi)} \in \mathsf{exc}(\mathcal{F})$, where $H_2 := R[\rho^{-1}([k])]$. Therefore, (G, k) is a yes-instance of $\mathcal{L}$-R-$\mathcal{F}$ if and only if there is an element $(\mathbf{R}, H_1, \phi, S_B)$ in the signature of $\mathbf{G}_r$ such that $(H_2, \phi) \in \mathcal{L}(H_1)$. This can be checked in time $2^{2\binom{k}{2} + \mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$.

We want to build the signature of $\mathbf{G}_t$, $t \in V(T)$, in a bottom-up fashion. Let $G_\emptyset$ be the graph with no vertices, and $\mathbf{G}_\emptyset$ be the corresponding boundaried graph. Let us assume that $\mathcal{F}$ does not contain $G_\emptyset$, since in that case the problem is trivial. Let Rep be the algorithm of Lemma 4.4.3.

Leaf nodes. Let t be a leaf of T. Given that $\beta(t) = \emptyset$, the signature of $\mathbf{G}_t$ is the singleton containing the tuple $(\mathbf{G}_\emptyset, G_\emptyset, \emptyset \to \emptyset, \emptyset)$. Constructing the signature for a leaf node takes time $\mathcal{O}(1)$.

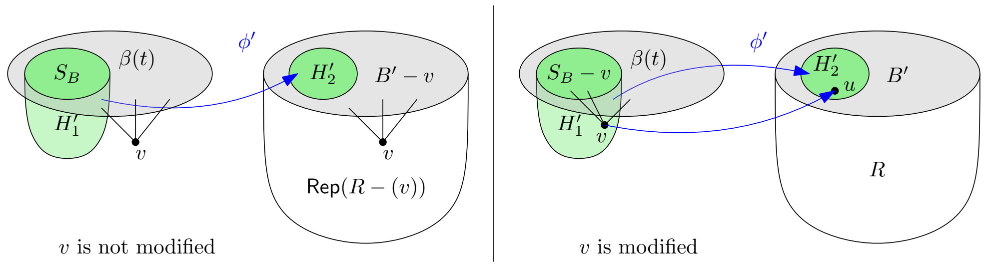

Figure 7.10: Forgetting a vertex v.

Forget nodes. When we forget a vertex v, we simply remove it from the boundary of $\mathbf{R}$ if it is not in the partial solution, i.e., in S_B. Otherwise, if it is in the partial solution, we still need to remember it, so it remains in the boundary of $\mathbf{R}$. However, we remove it from S_B as it does not belong to the current bag anymore. See Figure 7.10 for an illustration. More formally, we do as follows.

Let $t \in V(T)$ be a forget node of T. Let t' be the child of t and $v \in \beta(t') \setminus \beta(t)$ be the forgotten vertex. The signature of $\mathbf{G}_t$ is the set constructed by adding, for each tuple $(\mathbf{R} = (R, B', \rho'), H'_1, \phi', S_B)$ of the signature of $\mathbf{G}_{t'}$, the following tuple:

- *(v is not part of the modification)* if $v \in S_B$, then $(\mathsf{Rep}(\mathbf{R} - (v)), H'_1, \phi', S_B)$, where $\mathbf{R} - (v) := (R, B' \setminus \{v\}, \rho'|_{B' \setminus \{v\}})$,
- *(v is part of the modification)* otherwise, $(\mathbf{R}, H'_1, \phi', S_B \setminus \{v\})$.

Given that $\mathbf{R} \in \mathcal{R} \subseteq \mathcal{R}^{k+w}_{\ell_\mathcal{F}}$ and that R does not contain $K_{s_\mathcal{F}}$ as a minor, by Proposition 4.4.1, $|V(R)| = \mathcal{O}_{\ell_\mathcal{F}}(k+w)$. Thus, by Lemma 4.4.3, $\mathsf{Rep}(\mathbf{R}-(v))$ can be computed in time $2^{\mathcal{O}_{\ell_\mathcal{F}}((k+w)\log(k+w))}$. Given that the signature of $\mathbf{G}_{t'}$ has at most $2^{k^2+\mathcal{O}_{\ell_\mathcal{F}}((k+w)\log(k+w))}$ elements, we conclude that constructing the signature of $\mathbf{G}_t$ takes time $2^{k^2+\mathcal{O}_{\ell_\mathcal{F}}((k+w)\log(k+w))}$.

Introduce nodes. When we introduce a vertex v, we guess whether v belongs to S' or not, and if we guess that it does, we also guess how it is modified: it can either be deleted ($\phi'(v) = \emptyset$), or identified to a vertex ($\phi'(v) = \phi'(u)$ for some $u \in V(G_t) \setminus \{v\}$), or it can be a new vertex in H'_2 (when $\phi'^{-1}(\phi'(v)) = \{v\}$). In this latter case, we also need to guess the edges between $\phi'(v)$ and $u \in V(H'_2) \setminus \{\phi'(v)\}$ to get H'_2. If v is not part of the modification, then we simply add v to $\mathbf{R}$ (and to its boundary given that v is in the current bag). If v is modified, then it is added to H'_1, and we need to check that the obtained graph $H'_1 + v$ has at most k vertices. In the case when v is either deleted or identified to another vertex, H'_2 does not change, but v must be additionally mapped by ϕ' to $\emptyset$ in the first case, or to a modified vertex, that is, a vertex in $\rho^{-1}([k])$, in the second case. When v is deleted, $\mathbf{R}$ does not change either, but when v is identified, we need to add the edges between $\phi'(v)$ and the vertices adjacent to v in G_t. Otherwise, v is mapped to a new vertex, in which case we add a new vertex to H'_2, and by extension, to $\mathbf{R}$, we guess its adjacencies to H'_2, and we guess its label in $[k]$. In any case, we need to check that the modified graph is indeed in $\mathsf{exc}(\mathcal{F})$. See Figure 7.11 for an illustration. More formally, we do as follows.

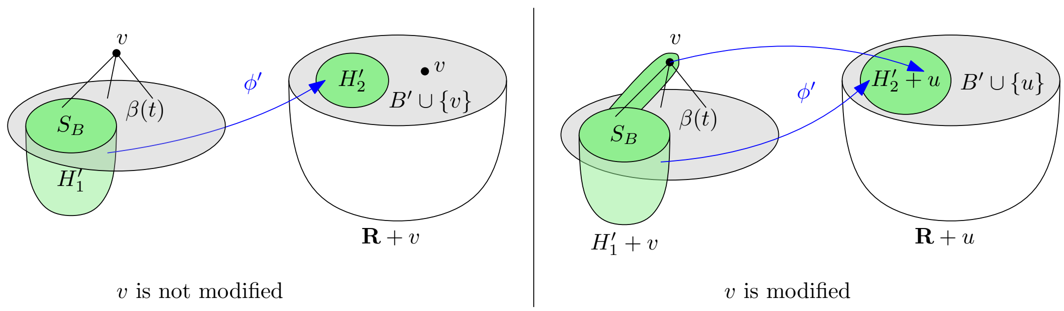

Figure 7.11: Introducing a vertex v.

Let $t \in V(T)$ be an introduce node of T. Let t' be the child of t and $v \in \beta(t) \setminus \beta(t')$ be the introduced vertex. Let $E_t \subseteq E(G[\beta(t)])$ be the set of edges uv for $u \in \beta(t)$. The signature of $\mathbf{G}_t$ is the set constructed by adding, for each tuple $(\mathbf{R} = (R, B', \rho'), H_1', \phi', S_B)$ of the signature of $\mathbf{G}_{t'}$, the following tuples:

- *(v is not part of the modification)* if $\mathbf{R} + v \in \mathsf{exc}(\mathcal{F})$, then $(\mathsf{Rep}(\mathbf{R}+v), H_1', \phi', S_B)$, where
  - $\mathbf{R} + v := (R + v, B' \cup \{v\}, \rho' \cup (v \mapsto \rho_t(v))$ and
  - $R + v$ is the graph with vertex set $V(R) \cup \{v\}$ and edge set the union of $E(R)$ and, for each edge $uv \in E_t$, the edge uv if $u \notin S_B$ or the edge $\phi'(u)v$ if $u \in S_B$ and $\phi'(u) \neq 0$,
- *(v is deleted)* if $|V(H_1')| \leq k-1$, then $(\mathbf{R}, H_1' + v, \phi' \cup (v \mapsto \emptyset), S_B \cup \{v\})$, where
  - $H_1' + v$ is the graph with vertex set $V(H_1') \cup \{v\}$ and edge set the union of $E(H_1')$ and the edges $uv \in E_t$ for each $u \in S_B$,
- *(v is identified to a vertex u that is in the partial solution)* if $|V(H_1')| \leq k-1$, then, for each $u \in \rho^{-1}([k])$, $(\mathsf{Rep}(\mathbf{R}'), H_1' + v, \phi' \cup (v \mapsto u), S_B \cup \{v\})$, where
  - $\mathbf{R}'$ is obtained from $\mathbf{R}$ by adding an edge uw for each $w \in \beta(t) \setminus S_B$ such that $vw \in E(G)$ and
  - $H_1' + v$ is defined as above.
- *(v is part of the modification but not deleted nor identified to another vertex in the partial solution)* for each $i \in [k]$ such that $\rho'^{-1}(i) = \emptyset$, for each $(H_2' + u_i, \phi' \cup (v \to u_i)) \in \mathcal{M}(H_1' + v)$ whose restriction to $V(H_1')$ is (H_2', ϕ'), if $|V(H_1')| \leq k-1$ and $\mathbf{R} + u_i \in \mathsf{exc}(\mathcal{F})$, then $(\mathsf{Rep}(\mathbf{R} + u_i), H_1' + v, \phi' \cup (v \mapsto u_i), S_B \cup \{v\})$, where
  - $\mathbf{R} + u_i := (R + u_i, B' \cup \{u_i\}, \rho' \cup (u_i \mapsto i))$,
  - $R + u_i$ is the graph with vertex set $V(R) \cup \{u_i\}$ and edge set the union of $E(R)$ and the edges uu_i for each edge $uu_i \in E(H_2' + u_i)$,
  - $H_2' := R[(\phi')^+(V(H_1'))]$, and
  - $H_1' + v$ is defined as above.

As proved in the forget case, $|V(R)| = \mathcal{O}_{\ell_{\mathcal{F}}}(k+w)$. Therefore, by [205], checking whether $\mathbf{R}+v \in \mathsf{exc}(\mathcal{F})$ takes time $\mathcal{O}_{\ell_{\mathcal{F}}}((k+w)^{1+o(1)})$. And again, by Lemma 4.4.3, $\mathsf{Rep}(\mathbf{R}+v)$ can be computed in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$. Given that $|V(H_2')| \leq |V(H_1')| \leq k-1$, there are at most 2^{k-1} choices for $H_2' + u_i$. Hence, given that the signature of $\mathbf{G}_{t'}$ has at most $2^{k^2+\mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$ elements, we conclude that constructing the signature of $\mathbf{G}_t$ takes time $2^{k^2+\mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$.

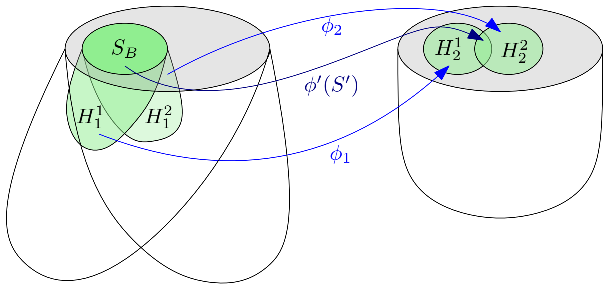

Figure 7.12: Joining bags.

Join nodes. When we join two bags, we join any partial solutions of both sides that are compatible together. They are compatible if the restriction to the current bag is the same on both sides. Vertices on the same label in $[k]$ are vertices of the modification that are identified. Additionally, we need to guess the edges that may be added between vertices of $H_2^1 - V(H_2^2)$ and vertices of $H_2^2 - V(H_2^1)$. See Figure 7.12 for an illustration. More formally, we do as follows.

Let $t \in V(T)$ be a join node of T. Let t_1 and t_2 be the children of t. The signature of $\mathbf{G}_t$ is the set constructed by adding, for each tuple $(\mathbf{R_i} = (R_i, B_i, \rho_i), H_1^i, \phi_i, S_B)$ of the signature of $\mathbf{G}_{t_i}$, for $i \in [2]$, each tuple $(\mathsf{Rep}(\mathbf{R}'), H_1, \phi, S_B)$ such that:

- $|V(H_1)| \leq k$ and $(H_2', \phi) \in \mathcal{M}(H_1)$, where:
  - $H_1 := \mathbf{H}_1^1 \oplus \mathbf{H}_1^2$, where $\mathbf{H}_1^i := (H_1^i, S_B, \rho_t|_{S_B})$ for $i \in [2]$,
  - $H_2 := \mathbf{H}_2^1 \oplus \mathbf{H}_2^2$, where $H_2^i := R_i[\rho_i^{-1}([k])]$ and $\mathbf{H}_2^i := (H_2^i, \phi_i^+(S_B), \rho_i|_{\phi_i(S_B)})$ for $i \in [2]$,
  - H_2' is obtained from H_2 by adding a set E' of edges between vertices of $H_2^1 - \phi_1(S_B)$ and vertices of $H_2^2 - \phi_2(S_B)$, and
  - $\phi_1|_{S'} = \phi_2|_{S'}$, allowing us to define that $\phi : V(H_1) \to V(H_2)$ such that $\phi|_{V(H_1^i)} = \phi_i$ for $i \in [2]$,
- $\mathbf{R}' \in \mathsf{exc}(\mathcal{F})$, where
  - $\mathbf{R}'$ is obtained from $(\mathbf{R}_1' \oplus \mathbf{R}_2', B, \rho) \in \mathsf{exc}(\mathcal{F})$ by adding the edge set E' to the underlying graph,
  - $\mathbf{R}_1'$ and $\mathbf{R}_2'$ are compatible, with $B_i' := B_i \setminus (\rho_i^{-1}([k]) \setminus \phi_i(S_B))$ and $\mathbf{R}_i' := (R_i, B_i', \rho_i|_{B_i'})$ (informally, $\mathbf{R}_i'$ is obtained from $\mathbf{R}_i$ by removing from the boundary the vertices that are part of the modification but not in the bag $\beta(t)$),
  - $B := B_1 \cup B_2$, which is well-defined given that $B_1 \cap B_2 = B_1' = B_2'$ by compatibility, and
  - ρ is a function such that $\rho|_{B_i} = \rho_i$ for $i \in [2]$, which is well-defined by compatibility.

The signature of each $\mathbf{G}_{t_i}$ has at most $2^{k^2+\mathcal{O}_{\ell_{\mathcal{F}}}((k+w)\log(k+w))}$ elements, and there are at most 2^{k^2} choices for E', so constructing the signature of $\mathbf{G}_t$ takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2+(k+w)\log(k+w))}$.

Running time. Given that each step takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2+(k+w)\log(k+w))}$ and the the tree decomposition has $\mathcal{O}(w+n)$ nodes by Proposition 4.3.3, we conclude that the dynamic programming algorithm takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2+(k+w)\log(k+w))} \cdot n$.

Remark that, while the dynamic programming algorithm solve here the decision problem, it suffices to apply standard backtracking to obtain a solution in case of a yes-instance.

CHAPTER 8

Elimination distance to minor-closedness

Contents

In this chapter, we prove the results presented in Section 2.4, which are restated here for convenience.

Theorem 2.4.1. *Let $\mathcal{H}$ be a minor-closed graph class. Then there is an algorithm that solves* Elimination Distance to $\mathcal{H}$ *in time* $2^{2^{2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}}}} \cdot n^2$.

If $\mathcal{H}$ is apex-minor-free, then this algorithm runs in time $2^{2^{\mathcal{O}_{s_{\mathcal{H}}}(k^2 \log k)}} \cdot n^2$.

Theorem 2.4.2. *Let $\mathcal{H}$ be an apex-minor-free graph class. Then there is an algorithm that solves* Elimination Distance to $\mathcal{H}$ *in time* $2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}} \cdot n^3$.

Theorem 2.4.3. *Let $\mathcal{H}$ be a minor-closed graph class. Then there is an algorithm that solves* Elimination Distance to $\mathcal{H}$ *in time* $2^{\mathcal{O}(k \cdot w + w \log w)} \cdot n$ *on graphs of treewidth at most w.*

Theorem 2.4.4. *Let $\mathcal{H}$ be a minor-closed graph class. Then the obstructions of $\mathcal{E}_k(\mathcal{H})$ have* $2^{2^{2^{2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}}}}}$ *vertices.*

Moreover, this bound drops to $2^{2^{k^{\mathcal{O}_{s_{\mathcal{H}}}(1)}}}$ *when $\mathcal{H}$ is apex-minor-free.*

More particularly, we sketch the algorithms in Section 8.1, we give additional definitions and notations in Section 8.2 and, we prove Theorem 2.4.1, Theorem 2.4.2, Theorem 2.4.3, and Theorem 2.4.4 in Section 8.3, Section 8.4, Section 8.5, and Section 8.6, respectively.

Some notations and conventions. By *k-elimination set* of a graph G for a graph class $\mathcal{H}$, we refer to a set $X \subseteq V(G)$ such that $\mathsf{td}(\mathsf{torso}(G, X)) \leq k$ and $G - X \in \mathcal{H}$. Additionally, we note $\mathsf{ed}_{\mathcal{H}} := \mathcal{H}\text{-}\mathsf{td}$. Recall that that $\mathcal{E}_k(\mathcal{H})$ is the class of graphs G such that $\mathsf{ed}_{\mathcal{H}}(G) \leq k$

As in the previous chapter, instead of considering a minor-closed graph class $\mathcal{H}$, we consider its obstruction set $\mathcal{F}$, and thus the minor-closed graph class $\mathsf{exc}(\mathcal{F})$. We define $a_{\mathcal{F}}$ as the minimum apex number of a graph in $\mathcal{F}$, we set $s_{\mathcal{F}} := \max\{|V(F)| \mid F \in \mathcal{F}\}$, and we define $\ell_{\mathcal{F}}$ to be the maximum detail of a graph in $\mathcal{F}$. Also, we say that $\mathcal{F}$ is *non-trivial* when all graphs in $\mathcal{F}$ contain at least two vertices.

8.1 Sketch of the algorithms

The strategy to solve Theorem 2.4.1 (cf. Theorem 8.3.1) and Theorem 2.4.2 (cf. Theorem 8.4.1) is similar to the one used in Subsection 7.3.2 to solve $\mathcal{L}$-Replacement to $\mathsf{exc}(\mathcal{F})$. It uses the same following ingredients.

Bounded treewidth: When the input graph as bounded treewidth, i.e. to prove Theorem 2.4.3, we design our own dynamic programming algorithm (cf. Section 8.5) combining the representative-based technique of [23] with the dynamic programming algorithm of [256] deciding whether the treedepth is at most k in FPT-time parameterized by $\mathsf{tw} + k$ (Proposition 8.5.4).

Irrelevant vertex: We use as a blackbox the irrelevant vertex technique from [285] (cf. Proposition 8.2.4), which is similar to the one of Chapter 7 (cf. Theorem 7.3.1).

Obligatory set: However, there is a difference due to the fact that the modulator has bounded treedepth instead of bounded size.

For the $\mathcal{L}$-Replacement to $\mathsf{exc}(\mathcal{F})$ problem, if we find a set A' that intersects any solution, then we can branch by guessing the intersection of A' with the modulator and recursively solving the reduced instance obtained after doing the guessed modification. As proved in Subsection 7.3.2, this step is applied $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2)}$ times. This bound comes from the fact that we know that the modulator has at most k vertices, and that we guess at least one vertex of the modulator each time we enter this step.

However, for Elimination Distance to $\mathsf{exc}(\mathcal{F})$, the size of the modulator (a k-elimination set) may not depend on k. Thus, the number of time this step may be applied depends exponentially on n (instead of k), which does not give an FPT-algorithm. To circumvent this problem, we propose two alternatives:

Option 1: The first alternative is to not use the branching technique (Step 5 in Subsection 7.3.2), but only the irrelevant vertex technique. In this case, when using the algorithms `Grasped-or-Flat` (Proposition 4.6.2) and `Clique-or-twFlat` (Proposition 4.6.3), we force the outcome to be an apex set A and a flatness pair of $G - A$, using the fact that $(K_{s_{\mathcal{F}}+k}, k)$ is a no-instance of the problem. However, the bound on the size of A now depends on k, and thus, so does the variable a in the input of the algorithm `Homogeneous` (Proposition 4.6.12). This explains the triple-exponential parametric dependence on k in Theorem 2.4.1. Interestingly, a precise analysis of the time complexity, which can be found in Section 8.3, shows that if $a_{\mathcal{F}} = 1$, i.e., when $\mathcal{F}$ contains an apex graph, the parametric dependence is only double-exponential on k (cf. Theorem 2.4.1).

Option 2: The second alternative is to restrict ourselves to the case where $a_{\mathcal{F}} = 1$. Thus, in the branching case (Step 5 in Subsection 7.3.2), we find a vertex v that belongs to every k-elimination set. There is no need to branch, and this step is done at most n times. However, the fact that the time complexity of this step is quadratic in n explains the cubic complexity of the algorithm in Theorem 2.4.2.

8.2 Preliminaries

In this section, we give some more definitions. Namely, in Subsection 8.2.1, we define elimination trees, that is an alternative way to define elimination distance of a graph to a graph class and, in Subsection 8.2.2, we define the notion of bidimensionality with respect to a wall, which is crucial to be able to apply the irrelevant vertex technique for Elimination Distance to $\mathsf{exc}(\mathcal{F})$.

8.2.1 $\mathcal{F}$-elimination trees

We start this subsection by defining some notions on (rooted) trees.

Notations on trees and rooted trees. Let T be a tree and u, v be two nodes of T. We denote by uTv the path in T between u and v. A *rooted tree* is a pair (T, r) where T is a tree and r is a node of T called *root* of (T, r). Let (T, r) be a rooted tree and let u be a node of T. We define the *descendants* of u in (T, r) by $\mathsf{Desc}_{T,r}(u) = \{x \in V(T) \mid u \in V(xTr)\}$ and the *ancestors* of u in (T, r) by $\mathsf{Anc}_{T,r}(u) = V(rTu) \setminus \{u\}$. We define the *leaves* in (T, r) by $\mathsf{Leaf}(T, r) = \{u \in V(T) \mid \mathsf{Desc}_{T,r}(u) = \{u\}\}$ and the *internal nodes* in (T, r) by $\mathsf{Int}(T, r) = V(T) \setminus \mathsf{Leaf}(T, r)$. If $u \neq r$ then we denote by $\mathsf{Par}_{T,r}(u)$ the unique node in $\mathsf{Anc}_{T,r}(u) \cap N_T(u)$. We also agree that $\mathsf{Par}_{T,r}(r) = \mathsf{void}$. We denote by $\mathsf{Ch}_{T,r}(u) = \mathsf{Desc}_{T,r}(u) \cap N_G(u)$ the set of the children of u (certainly $\mathsf{Ch}_{T,r}(u) = \emptyset$ if u is a leaf of T). Given $K \subseteq V(T)$, the *least common ancestor* of K in (T, r) is the node u such that $K \subseteq \mathsf{Desc}_{T,r}(u)$ and there is no child v of u such that $K \subseteq \mathsf{Desc}_{T,r}(v)$.

The *height function* $\mathsf{height}_{T,r} : V(T) \to \mathbb{N}$ maps $v \in \mathsf{Leaf}(T, r)$ to 0 and $v \in \mathsf{Int}(T, r)$ to $1 + \max\{\mathsf{height}_{T,r}(x) \mid x \in \mathsf{Ch}_{T,r}(v)\}$. The *height* of (T, r) is $\mathsf{height}_{T,r}(r)$. Note that the height function is decreased by one here compared to the usual definition of the height.

We use (T_u^r, u) to denote the rooted tree where $T_u^r = T[\mathsf{Desc}_{T,r}(u)]$ and we call (T_u^r, u) *subtree of* (T, r) *rooted at* u. To simplify notation and when the root r is clear from the context, we use T_u instead of T_u^r.

A *rooted forest* is a pair (F, R) where F is a forest and R is a set of roots such that each tree in F has exactly one root in R. All notations above naturally extend to forests.

Elimination trees. We now define *elimination trees*, that can be used to define alternatively graphs of bounded elimination distance. Let $\mathcal{F}$ be a non-empty finite collection of non-empty graphs. An $\mathcal{F}$-*elimination tree* of a connected graph G is a triple (T, χ, r) where (T, r) is a rooted tree and $\chi : V(T) \to 2^{V(G)}$ such that:

- for each $t \in \mathsf{Int}(T, r)$, $|\chi(t)| = 1$,
- $(\chi(t))_{t \in V(T)}$ is a partition of $V(G)$,
- for each $uv \in E(G)$, if $u \in \chi(t_1)$ and $v \in \chi(t_2)$, then $t_1 \in \mathsf{Anc}_{T,r}(t_2) \cup \mathsf{Desc}_{T,r}(t_2)$,
- for each $t \in \mathsf{Leaf}(T, r)$, $G[\chi(t)] \in \mathsf{exc}(\mathcal{F})$, and
- for each $t \in V(T)$, $G[\chi(T_t)]$ is connected.

The *height* of (T, χ, r) is the height of (T, r). It is straightforward to see that the minimum height of an $\mathcal{F}$-elimination tree of a connected graph G is $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G)$. Note that $\chi(\mathsf{Int}(T, r))$ is a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$ and that, if $\mathcal{F}$ is trivial, then, for each $t \in \mathsf{Leaf}(T, r)$, $\chi(t) = \emptyset$. Observe also that for every $u \in \mathsf{Int}(T, r)$ with at least two children x and y, any path between $\chi(T_x)$ and $\chi(T_y)$ intersects $\chi(uTr)$.

An $\mathcal{F}$-*elimination forest* of a graph G is a triple (F, χ, R), such that, if $\mathsf{cc}(G) = \{G_1, ..., G_l\}$, then F is the disjoint union of the trees $T_1, \dots, T_l$ and $R = \{r_1, ..., r_l\}$ where $(T_i, \chi|_{V(T_i)}, r_i)$ is an $\mathcal{F}$-elimination tree of G_i for $i \in [l]$.

The following simple lemma is based on the fact that, given an $\mathcal{F}$-elimination tree (T, χ, r) of a graph G, for every non-leaf node u of T, $\chi(uTr)$ separates the vertex sets $\chi(T_x)$ and $\chi(T_y)$, where x and y are distinct children of v in (T, r).

Lemma 8.2.1. *Let $\mathcal{F}$ be a finite collection of graphs. Let G be a graph and let H be a connected subgraph of G. Let (F, χ, R) be an $\mathcal{F}$-elimination forest of G. Then the least common ancestor of $\chi^{-1}(V(H))$ exists and belongs to $\chi^{-1}(V(H))$.*

Proof. Let $K := \chi^{-1}(V(H))$. Since H is connected, K is a subset of a tree in F, and therefore the least common ancestor of K is defined. Let u be the least common ancestor of K. Let $r \in R$ be the root of the tree containing u. Let $x, y \in V(H)$ such that the least common ancestor of $\chi^{-1}(x)$ and $\chi^{-1}(y)$ is u. Since H is connected, there is a path P in H between x and y. By the third property of elimination trees, $\{u\} \cup \mathsf{Anc}_{F,R}(u)$ intersects $\chi^{-1}(V(P))$, and so $\{u\} \cup \mathsf{Anc}_{F,R}(u)$ intersects K. Since u is the least common ancestor of K, $u \in K$. $\square$

We now present a lemma to justify that the graphs with bounded elimination distance to $\mathsf{exc}(\mathcal{F})$ are minor-free. Intuitively, the proof of this lemma is based on the fact that, due to Lemma 8.2.1, the size of the largest clique minor that can "fit" inside an elimination tree is equal to the height of the elimination tree.

Lemma 8.2.2. *Let $\mathcal{F}$ be a finite collection of graphs. Let G be a graph and $k \in \mathbb{N}$ such that $\mathsf{ed}_{\mathsf{exc}\mathcal{F}}(G) \le k$. Then $K_{s_{\mathcal{F}}+k}$ is not a minor of G.*

Proof. Let (F, χ, R) be an $\mathcal{F}$-elimination forest of G of height at most k. Suppose towards a contradiction that there is a model of $K_{s_{\mathcal{F}}+k}$ in G. Let $x_1, ..., x_{s_{\mathcal{F}}+k}$ be the vertices of $K_{s_{\mathcal{F}}+k}$ and for every $i \in [s_{\mathcal{F}} + k]$, let V_i be the model of x_i in G. Let G' be the graph obtained by contracting, for each $i \in [s_{\mathcal{F}} + k]$, the edges in each V_i. Let $v_i, i \in [s_{\mathcal{F}} + k]$ be the resulting vertices after the contraction of each V_i. Thus, the graph $G'[\{v_1, ..., v_{s_{\mathcal{F}}+k}\}]$ is isomorphic to $K_{s_{\mathcal{F}}+k}$.

Let (F', χ', R) be obtained from (F, χ, R) as follows. For every $i \in [s_{\mathcal{F}} + k]$, let u_i be the least common ancestor of $K_i := \chi^{-1}(V_i)$ in (F, R). Due to Lemma 8.2.1, $u_i \in K_i$. The forest F' is obtained after removing each node $v \in (K_i \setminus u_i) \cap (\mathsf{Int}(F, R) \setminus R)$ from $V(F)$ and adding an edge between $\mathsf{Par}_{F,R}(v)$ and each node in $\mathsf{Ch}_{F,R}(v)$. The function χ' is defined as $\chi'(v) := \chi(v)$ if $v \in \mathsf{Int}(F', R)$ and $\chi'(v) := \chi(v) \setminus (V_i \setminus u_i)$ if $v \in \mathsf{Leaf}(F', R)$. In the latter case, if $G'[\chi'(v)]$ is not connected, then we update F' by replacing v by $|\mathsf{cc}(G'[\chi'(v)])|$ nodes, each one associated with a connected component of $G'[\chi'(v)]$. Observe that (F', χ', R) is an $\mathcal{F}$-elimination forest of G' of height at most k and that we can assume that for every $i \in [s_{\mathcal{F}} + k]$, $v_i \in \chi'(u_i)$.

Since the vertices in $\{v_1, ..., v_{s_{\mathcal{F}}+k}\}$ are pairwise connected by an edge in G', the third property of elimination trees implies that there is $u \in \mathsf{Leaf}(F', R)$ and $r \in R$ such that $\{v_1, ..., v_{s_{\mathcal{F}}+k}\} \subseteq \chi'(uF'r)$. Let $u' := \mathsf{Par}_{F',R}(u)$. For every $t \in u'F'r$, $t \in \mathsf{Int}(F', R)$, so $|\chi'(t)| = 1$, and therefore, $|\chi'(u'Fr)| \leq k$. Thus, $|\chi'(u) \cap V(H)| \geq s_{\mathcal{F}}$. Therefore, $K_{s_{\mathcal{F}}}$ is a minor of $G'[\chi'(u)]$. This contradicts the fact that $G'[\chi'(u)] \in \mathsf{exc}(\mathcal{F})$, so $K_{s_{\mathcal{F}}+k}$ is not a minor of G. □

We also show that given a graph G and an integer k, the removal of a k-elimination set from G does not decrease the treewidth of G more than k.

Lemma 8.2.3. *Let $\mathcal{F}$ be a finite collection of graphs. Let c, k be two integers and let G be a graph such that $\mathsf{tw}(G) \geq c$. Let S be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. Then $\mathsf{tw}(G - S) \geq c - k$.*

Proof. Suppose first that G is connected. Let (T, χ, r) be an $\mathcal{F}$-elimination tree of G with $\chi(\mathsf{Int}(T, r)) = S$. Let $v_1, ..., v_l$ be the leaves of (T, r), whose label is given by a depth-first search order starting from r. Let $C_i := G[\chi(v_i)]$ for $i \in [l]$, and note that $\mathsf{tw}(G - S) = \max_{i \in [l]} \mathsf{tw}(C_i)$. Suppose for contradiction that $\mathsf{tw}(G - S) < c - k$, and we proceed to show that $\mathsf{tw}(G) < c$, contradicting our hypothesis. Let (T_i, β_i) be an optimal tree decomposition of C_i of width w_i and let P_i be the path from the parent of v_i to r in T, for $i \in [l]$. Let $w := \max_{i \in [l]} w_i$, so we have that $w \leq c - k - 1$. We construct a tree decomposition $(\mathcal{T}, \beta)$ of G, starting from the tree decompositions (T_i, β_i), as follows. Create a path with nodes $x_1, ..., x_l$ such that for $i \in [l]$, $\beta(x_i) = V(P_i)$. Then for $i \in [l]$, add an edge between x_i and a node of T_i. For each $x \in V(T_i)$, we set $\beta(x) := \beta_i(x) \cup V(P_i)$. Since the height of (T, r) is at most k, P_i has size at most k for $i \in [l]$, so $(\mathcal{T}, \beta)$ has width at most $w + k \leq c - 1$, a contradiction.

If G is not connected, we can apply the above proof to each of its connected components. □

8.2.2 Bidimensionality of elimination sets

In this subsection, we present the notion of bidimensionality of a set X with respect to a wall W of a graph G. Essentially, if X has big bidimensionality with respect to W, then X has big bidimensionality $\mathsf{bg}(G, X)$ (see Section 1.6).

Bidimensionality. Let W be a wall of a graph G, $\tilde{\mathcal{Q}}$ be a W-canonical partition of G, and $X \subseteq V(G)$. The *bidimensionality* of X in G with respect to $\tilde{\mathcal{Q}}$, denoted by $\mathsf{bid}_{\tilde{\mathcal{Q}}}(X)$, is the number of internal bags of $\tilde{\mathcal{Q}}$ intersected by X. The *bidimensionality* of X in G with respect to W, denoted by $\mathsf{bid}_{G,W}(X)$, is the maximum bidimensionality of X with respect to a W-canonical partition of G.

As discussed in Section 1.6, the irrelevant vertex technique holds for modulators of bounded bidimensionality. The combinatorial version of this result is stated in [285, Lemma 16] and can be algorithmized using [286, Theorem 5] (Proposition 4.6.6). Recall that the Unique Linkage theorem was mentioned in Subsection 4.6.5.

Proposition 8.2.4 ([285, 286]). *Let $\mathcal{F}$ be a finite collection of graphs. There exist two functions $f_{8.2.4} : \mathbb{N}^4 \to \mathbb{N}$ and $g_{8.2.4} : \mathbb{N}^2 \to \mathbb{N}$, and an algorithm with the following specifications:*

`Find-Irrelevant-Vertex`$(k, a, G, A, W, \mathcal{R})$
Input: *Two integers $k, a \in \mathbb{N}$, a graph G, a set $A \subseteq V(G)$, and a regular and tight flatness pair $(W, \mathcal{R})$ of $G - A$ of height at least $f_{8.2.4}(a, \ell_{\mathcal{F}}, 3, k)$ that is $g_{8.2.4}(a, \ell_{\mathcal{F}})$-homogeneous with respect to $\binom{A}{\leq a}$.*
Output: *A vertex v of $G - A$ such that for every set $X \subseteq V(G)$ with $\mathsf{bid}_{G-A,W}(X) \leq k$ and $|A \setminus X| \leq a$, it holds that $G - X \in \mathsf{exc}(\mathcal{F})$ if and only if $G - (X \setminus v) \in \mathsf{exc}(\mathcal{F})$.*
Moreover, $f_{8.2.4}(a, \ell_{\mathcal{F}}, q, k) = \mathcal{O}(k \cdot (f_{\mathsf{ul}}(16a + 12\ell_{\mathcal{F}}))^3 + q)$, where f_{ul} is the function of the Unique Linkage Theorem and $g_{8.2.4}(a, \ell_{\mathcal{F}}) = a + \ell_{\mathcal{F}} + 3$, and this algorithm runs in time $\mathcal{O}(n + m)$.

In this case, similarly to Theorem 7.3.1 in Chapter 7, we use the following result of [285] that basically says that if there is a big enough flat wall W and an apex set A' of $a_{\mathcal{F}}$ vertices that are all adjacent to many bags of a canonical partition of W, then each k-elimination set intersects A'. See Figure 7.3 for an illustration.

Proposition 8.2.5 ([285]). *There exist three functions $f_{8.2.5}, g_{8.2.5}, h_{8.2.5} : \mathbb{N}^3 \to \mathbb{N}$, such that if G is a graph, $k \in \mathbb{N}$, A is a subset of $V(G)$, $(W, \mathfrak{R})$ is a flatness pair of $G - A$ of height at least $f_{8.2.5}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$, $\tilde{\mathcal{Q}}$ is a W-canonical partition of $G - A$, A' is a subset of vertices of A that are adjacent, in G, to vertices of at least $g_{8.2.5}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$ $h_{8.2.5}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$-internal bags of $\tilde{\mathcal{Q}}$, and $|A'| \geq a_{\mathcal{F}}$, then for every set $X \subseteq V(G)$ such that $G - X \in \mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-A,W}(X) \leq k$, it holds that $X \cap A' \neq \emptyset$. Moreover, $f_{8.2.5}(a, s, k) = \mathcal{O}(2^a \cdot s^{5/2} \cdot k^{5/2})$, $g_{8.2.5}(a, s, k) = \mathcal{O}(2^a \cdot s^3 \cdot k^3)$, and $h_{8.2.5}(a, s, k) = \mathcal{O}((a^2 + k) \cdot s)$, where $a = a_{\mathcal{F}}$ and $s = s_{\mathcal{F}}$.*

Lemma 8.2.7 provides a set $X \subseteq V(G)$ that can be used to apply Proposition 8.2.5 to Elimination Distance to $\mathsf{exc}(\mathcal{F})$. Given a k-elimination set S, we can find a set $X \supseteq S$ of bidimensionality at most $k(k+1)/2$ such that $G - X \in \mathsf{exc}(\mathcal{F})$. To prove this, we first prove the following result, which intuitively states that a k-elimination set can intersect at most k horizontal and vertical paths of a wall.

Lemma 8.2.6. *Let $\mathcal{F}$ be a finite collection of graphs. Let G be a graph, let $k \in \mathbb{N}$, let r, h be odd integers with $r \geq h + k$, let W be an r-wall of G, and let S be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. Then there is an h-subwall W' of W with $V(W') \cap S = \emptyset$.*

Proof. Since S is a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$, there is an $\mathcal{F}$-elimination forest (F, χ, R) of G of height k such that $\chi(\mathsf{Int}(F, R)) = S$.

We set $W_0 := W$. For $i \in [k]$, we proceed to construct an r_i-subwall W_i of the r_{i-1}-wall W_{i-1} with $r_i \geq r - i$ such that for every $i \in [k]$ there is a node z_i of F such that (F_{z_i}, z_i) has height at most $k - i$ and $V(W_i) \subseteq \chi(V(F_{z_i}))$. This will imply the existence of a wall of size at least $r - k$ whose vertex set will be a subset of $\chi(V(F_z)$, where $z \in \mathsf{Leaf}(F, R)$.

Let $S_i := V(W_{i-1}) \cap S$. If $S_i = \emptyset$, we set $W_j := W_{i-1}$ for $j \in [i, k]$. Otherwise, let u_i be the least common ancestor of $\chi^{-1}(S_i)$ in (F, R). According to Lemma 8.2.1, since W_{i-1} is connected, u_i exists and belongs to $\chi^{-1}(S_i)$.

We obtain an $(r_{i-1} - 1)$-subwall W_i of W_{i-1} that does not contain $\chi(u_i)$ by taking the wall containing all horizontal and vertical paths of W_{i-1} aside from the ones intersecting u_i and we set

$r_i := r_{i-1} - 1$ (to simplify the argument, here we call the resulting graph a wall even when the height is even). Note that since W_i is a subgraph of $G[\chi(V(F_{u_i}))]$ that is connected, there is a $z_i \in \mathsf{Ch}_{F_{u_i},u_i}(u_i)$ such that $V(W_i) \subseteq V(F_{z_i})$. Notice that $(F_{u_i}, \chi|_{V(F_{u_i})}, u_i)$ is an $\mathcal{F}$-elimination forest of $G[\chi(F_{u_i})]$ of height at most $k - i$.

Observe that z_k should be a leaf of (F, R) and therefore, since we have that $V(W_k) \subseteq V(F_{z_k}) \subseteq V(G) \setminus S$, W_k is a wall of $G - S$ of height at least $r - k \geq h$. □

Now we prove our result regarding the bidimensionality of k-elimination sets.

Lemma 8.2.7. *Let $\mathcal{F}$ be a finite collection of graphs. Let G be a graph, let $A \subseteq V(G)$, let $k \in \mathbb{N}$, let $r \geq 2k+3$ be an odd integer, let $(W, \mathfrak{R})$ be a flatness pair of $G - A$, and let S be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. There is a set $X \supseteq S$ such that $G - X \in \mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-A,W}(X) \leq k(k+1)/2$.*

Proof. Let $p = \mathsf{odd}(r - k)$. Let W' be a p-subwall of W that is a wall of $G - S$, which exists due to Lemma 8.2.6. Let C be the connected component of $G - S$ that contains W'. Since $C \in \mathsf{exc}(\mathcal{F})$, $K_{s_\mathcal{F}}$ is not a minor of C. Moreover, since S is a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$, there is a set $P \subseteq S$ of size at most k such that $(L, R) := (V(G) \setminus V(C), V(C) \cup P)$ is a separation of G with $L \cap R = P$.

Let us show that $\mathsf{bid}_{G-A,W}(V(G) \setminus V(C)) \leq k(k+1)/2$. Let $\tilde{\mathcal{Q}}$ be a W-canonical partition of $G - A$. Let l be the number of internal bags of $\tilde{\mathcal{Q}}$ intersected by P and note that $l \leq k$.

Let G' be the graph obtained from G after contracting each bag of $\tilde{\mathcal{Q}}$ to a vertex. It is easy to observe that G' is isomorphic to a planar supergraph of an h-grid H, where $h = r - 2$, together with an additional vertex that is adjacent to every vertex of the perimeter of H.

We let $[h]^2$ be the vertex set of H, where (i, j) and (i', j') are adjacent if and only if $|i-i'|+|j-j'| = 1$. We will show that there is a separation (L', R') of H of order at most l that maximizes $\min\{|L'|, |R'|\}$. Let $A = L' \cap R'$. We suppose without loss of generality that $|L'| \leq |R'|$. Notice that $l < h$. We take $A := \{(i, j) \in [h]^2 \mid i + j = l\}$, i.e., L' is the set of pairs of indices in the triangle bounded by $(0, 0)$, $(0, l)$, and $(l, 0)$. Thus, $|L'| = l(l + 1)/2$. It is easy to verify that this maximizes $|L'|$.

Therefore, since the vertices of H are the internal bags of $\tilde{\mathcal{Q}}$ and P intersects l internal bags, it implies that one of L and R intersects at most $l(l + 1)/2 \leq k(k + 1)/2$ internal bags of $\tilde{\mathcal{Q}}$. Recall that W' is a wall of C of height p. It is easy to verify that an elementary x-wall W^* has $2x^2 - 2$ vertices with $8x - 10$ vertices in the perimeter. Hence, it has $2(x - 2)^2$ vertices not in the perimeter, and therefore the canonical partition of W^* has $(x - 2)^2$ internal bags. Thus, the canonical partition of W' has $(p - 2)^2$ internal bags. Observe that each such a bag is contained in an internal bag of $\tilde{\mathcal{Q}}$ and therefore $V(C)$ intersects at least $(p - 2)^2$ internal bags of $\tilde{\mathcal{Q}}$. Since $(p - 2)^2 \geq (r - k - 2)^2 \geq (k + 1)^2 > k(k + 1)/2$, it holds that $\mathsf{bid}_{G-A,W}(V(C) \cup P) > k(k + 1)/2$. Therefore, $\mathsf{bid}_{G-A,W}(V(G) \setminus V(C)) \leq k(k + 1)/2$. □

8.3 Elimination distance to a minor-closed graph class

Let present our main result for Elimination Distance to $\mathsf{exc}(\mathcal{F})$. The following theorem is a restatement of Theorem 2.4.1.

Theorem 8.3.1. *For every finite collection of graphs $\mathcal{F}$, there exists an algorithm that, given a graph G and an integer k, decides whether $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \leq k$ in time $2^{2^{2^{k^{\mathcal{O}_{\ell_\mathcal{F}}(1)}}}} \cdot n^2$. In the particular case when $\mathcal{F}$ contains an apex-graph, this algorithm runs in time $2^{2^{\mathcal{O}_{\ell_\mathcal{F}}(k^2 \log k)}} \cdot n^2$.*

We will use the following result solving the problem on graphs of bounded treewidth. The proof of Theorem 8.3.2 is deferred to Section 8.5.

Theorem 8.3.2. *For every finite collection of graphs $\mathcal{F}$, there exists an algorithm that, given a graph G of treewidth at most* tw *and a non-negative integer k, decides whether $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \leq k$ in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{tw}\cdot k+\mathsf{tw}\log\mathsf{tw})} \cdot n$.*

In Subsection 8.3.1, we present an analogue of Proposition 7.1.2 (Lemma 8.3.3), which either reports an upper bound on the treewidth of the input graph, or finds a wall, or reports that we deal with a no-instance. The algorithm of Theorem 8.3.1 is described in Subsection 8.3.2 and, in Subsection 8.3.3, we present the proof of its correctness.

8.3.1 Quickly finding a wall

In this section, we prove the following result, that is the analog of Proposition 7.1.2 for Elimination Distance to $\mathsf{exc}(\mathcal{F})$.

Lemma 8.3.3. *Let $\mathcal{F}$ be a finite collection of graphs. There exist a function $f_{8.3.3} : \mathbb{N} \to \mathbb{N}$ and an algorithm with the following specifications:*

`Find-Wall`(G, r, k)
***Input**: A graph G, an odd $r \in \mathbb{N}_{\geq 3}$, and $k \in \mathbb{N}$.*
***Output**: One of the following:*

- **Case 1:** *Either a report that (G, k) is a* no*-instance of* Elimination Distance to $\mathsf{exc}(\mathcal{F})$*, or*
- **Case 2:** *a report that G has treewidth at most $f_{8.3.3}(s_{\mathcal{F}}) \cdot r + k$, or*
- **Case 3:** *an r-wall W of G.*

Moreover, $f_{8.3.3}(s_{\mathcal{F}}) = 2^{\mathcal{O}(s_{\mathcal{F}}^2 \cdot \log s_{\mathcal{F}})}$, and the algorithm runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r^2+k^2)} \cdot n$.

The proof is very similar to the one given in [284] for Proposition 7.1.2 in the case of Vertex Deletion to $\mathsf{exc}(\mathcal{F})$ and is achieved using the following result from Perkovic and Reed [250]. The main difference with respect to the proof of Proposition 7.1.2 given in [284] is that we need to use two new ingredients tailored for Elimination Distance to $\mathsf{exc}(\mathcal{F})$, namely Theorem 8.3.2 and Lemma 8.2.3.

Proposition 8.3.4 ([250])**.** *There exists an algorithm with the following specifications:*

***Input**: A graph G and $t \in \mathbb{N}$ such that $|V(G)| \geq 12t^3$.*
***Output**: A graph G^* such that $|V(G^*)| \leq (1 - \frac{1}{16t^2}) \cdot |V(G)|$ and,*

- *either G^* is a subgraph of G such that $\mathsf{tw}(G) = \mathsf{tw}(G^*)$, or*
- *G^* is obtained from G after contracting the edges of a matching in G.*

Moreover, the algorithm runs in time $2^{\mathcal{O}(t)} \cdot n$.

Proof of Lemma 8.3.3. Let $c := f_{4.6.1}(s_{\mathcal{F}}) \cdot r + k$.

Suppose that $|V(G)| < 12c^3$. Run the algorithm of [17] that, in time $\mathcal{O}(|V(G)|^{c+2}) = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}((r+k)\cdot\log(r+k))}$, checks whether $\mathsf{tw}(G) \leq c$. If this is the case, report the same and stop. If not, we aim to find an r-wall of G or conclude that we are dealing with a no-instance. First consider an arbitrary ordering $(v_1, \ldots, v_{|V(G)|})$ of the vertices of G. For each $i \in [|V(G)|]$, set G_i to be the graph induced by the vertices $v_1, \ldots, v_i$. Iteratively run the algorithm of Proposition 4.3.1 on G_i and c for increasing values of i. This algorithm runs in time $2^{\mathcal{O}(c)} \cdot |V(G)| = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r+k)}$. Let $j \in [|V(G)|]$ be the smallest integer such that the above algorithm outputs a report that $\mathsf{tw}(G_j) > c$ (it exists since $\mathsf{tw}(G) > c$) and notice that there exists a tree decomposition (T_j, β_j) of G_j of width at most $2c + 2$, obtained by the one of G_{j-1} by adding the vertex v_j to all the bags. Thus, we can call the algorithm of Theorem 8.3.2 with input $(G_j, 2c + 2, k)$, which runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(c\cdot(k+\log c))} \cdot |V(G_j)| = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}((r+k)\cdot(k+\log(r+k)))}$, in order to find, if it exists, a k-elimination set S_j of G_j for $\mathsf{exc}(\mathcal{F})$.

- If such a set S_j does not exist, then safely report that (G, k) is a no-instance.
- If such a set S_j exists, then call the algorithm of Proposition 4.3.4 for $G_j - S_j$, (and the decomposition of $G_j - S_j$ obtained from (T_j, β_j) by removing the vertices of S_j from all the bags) in order to check whether it contains an elementary r-wall W as a minor. This algorithm runs in time $2^{\mathcal{O}(c\cdot\log c)} \cdot r^{\mathcal{O}(c)} \cdot 2^{\mathcal{O}(r^2)} \cdot |V(G_j) \setminus S_j| = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}((r+k)\cdot\log(r+k))} \cdot r^{\mathcal{O}_{\ell_{\mathcal{F}}}(r+k)} \cdot 2^{\mathcal{O}(r^2)} = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r^2+(r+k)\cdot\log(r+k))}$, since $|E(W)| = \mathcal{O}(r^2)$. Since all connected components of $G_j - S_j$ are in $\mathsf{exc}(\mathcal{F})$, $G_j - S_j$ does not contain $K_{s_{\mathcal{F}}}$ as a minor. By Lemma 8.2.3, $\mathsf{tw}(G_j - S_j) \geq c - k = f_{4.6.1}(s_{\mathcal{F}}) \cdot r$. So because of Proposition 4.6.1, the algorithm of Proposition 4.3.4 will output an elementary r-wall W of $G_j - S_j$. We also return W as a wall of G.

Therefore, in the case where $|V(G)| < 12c^3$, we obtain one of the three possible outputs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r^2+k^2)}$.

If $|V(G)| \geq 12c^3$, then call the algorithm of Proposition 8.3.4 with input (G, c), which outputs a graph G^* such that $|V(G^*)| \leq (1 - \frac{1}{16c^2}) \cdot |V(G)|$ and

- either G^* is a subgraph of G such that $\mathsf{tw}(G) = \mathsf{tw}(G^*)$, or
- G^* is obtained from G after contracting the edges of a matching in G.

In both cases, recursively call the algorithm on G^* and distinguish the following two cases.

Case 1: G^* is a subgraph of G such that $\mathsf{tw}(G) = \mathsf{tw}(G^*)$.

(a) If the recursive call on G^* reports that $\mathsf{tw}(G^*) \leq c$, then return that $\mathsf{tw}(G) \leq c$.

(b) If the recursive call on G^* outputs an r-wall W of G^*, then return W as a wall of G.

(c) If (G^*, k) is a no-instance, then report that (G, k) is also a no-instance.

Case 2: G^* is obtained from G after contacting the edges of a matching in G.

(a) If the recursive call on G^* reports that $\mathsf{tw}(G^*) \leq c$, then do the following. First notice that the fact that $\mathsf{tw}(G^*) \leq c$ implies that $\mathsf{tw}(G) \leq 2c$, since we can obtain a tree decomposition (T, β) of G from a tree decomposition (T^*, β^*) of G^*, by replacing, in every $t \in V(T^*)$, every occurrence of a vertex of G^* that is a result of an edge contraction by its endpoints in G. Thus, we can call the algorithm of Theorem 8.3.2 with input $(G, 2c, k)$, which runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(c(k+\log c))} \cdot n$, in order to find, if it exists, a k-elimination set S of G for $\mathsf{exc}(\mathcal{F})$. We distinguish again two cases.

  - If such a set S does not exist, then the algorithm reports that (G, k) is a no-instance.

- If such a set S exists, then apply the algorithm of Proposition 4.3.1 with input $(G-S, 2c)$, which runs in time $2^{\mathcal{O}(c)} \cdot n$, and obtain a tree decomposition of $G-S$ of width at most $4c+1$. Using this decomposition, call the algorithm of Proposition 4.3.4 for $G-S$ in order to check whether it contains an elementary r-wall W as a minor. This algorithm runs in time $2^{\mathcal{O}(c\cdot\log c)} \cdot r^{\mathcal{O}(c)} \cdot 2^{\mathcal{O}(r^2)} \cdot n = 2^{\mathcal{O}_{\ell_\mathcal{F}}((r+k)\cdot\log(r+k))} \cdot r^{\mathcal{O}_{\ell_\mathcal{F}}(r+k)} \cdot 2^{\mathcal{O}(r^2)} \cdot n = 2^{\mathcal{O}_{\ell_\mathcal{F}}(r^2+(r+k)\cdot\log(r+k))} \cdot n$, since $|E(G-S)| = \mathcal{O}(n)$ and $|E(W)| = \mathcal{O}(r^2)$. If this algorithm outputs an elementary r-wall W of $G-S$, then output W. Otherwise, safely report, because of Proposition 4.6.1 and Lemma 8.2.3, that $\mathsf{tw}(G) \leq f_{4.6.1}(s_\mathcal{F}) \cdot r + k = c$.

(b) If the recursive call on G^* outputs an r-wall W^* of G^*, then by uncontracting the edges of M in W^*, we can return an r-wall of G.

(c) If (G^*, k) is a no-instance, then report that (G, k) is also a no-instance.

It is easy to see that the running time of the above algorithm is given by the function

$$T(n,k,r) \;\leq\; T\left((1-\frac{1}{16c^2})\cdot n, k, r\right) + 2^{\mathcal{O}_{\ell_\mathcal{F}}(r^2+k^2)} \cdot n,$$

which implies that $T(n,k,r) = 2^{\mathcal{O}_{\ell_\mathcal{F}}(r^2+k^2)} \cdot n$, as claimed. $\square$

8.3.2 Description of the algorithm for Elimination Distance to $\mathsf{exc}(\mathcal{F})$

Let us describe our algorithm for the general case.

We define the following constants.

$$\begin{aligned}
a &= g_{4.6.2}(s_\mathcal{F}+k), & q &= g_{8.2.5}(a_\mathcal{F}, s_\mathcal{F}, k(k+1)/2),\\
p &= h_{8.2.5}(a_\mathcal{F}, s_\mathcal{F}, k(k+1)/2), & l &= (q-1)\cdot a,\\
d &= g_{8.2.4}(a_\mathcal{F}-1, \ell_\mathcal{F}), & r_4 &= f_{8.2.4}(a_\mathcal{F}-1, \ell_\mathcal{F}, 3, k(k+1)/2),\\
r_3 &= f_{4.6.12}(r_4, a_\mathcal{F}-1, a, d), & r_2 &= \mathsf{odd}(\max\{f_{4.6.8}(l+1, r_3, p), f_{8.2.5}(a_\mathcal{F}, s_\mathcal{F}, k(k+1)/2)\}),\\
r_1 &= \mathsf{odd}(f_{4.6.3}(s_\mathcal{F}+k)\cdot r_2),
\end{aligned}$$

Note that $r_4 = \mathcal{O}_{\ell_\mathcal{F}}(k^2)$, $r_3 = \mathcal{O}_{\ell_\mathcal{F}}(k^{2\cdot c})$, $r_2 = \mathcal{O}_{\ell_\mathcal{F}}(k^{2\cdot c+15})$, and $r_1 = 2^{\mathcal{O}_{\ell_\mathcal{F}}(k^2\log k + c\log k)}$, where $c = g_{4.6.12}(a_\mathcal{F}-1, a, d) = 2^{\mathcal{O}_{\ell_\mathcal{F}}(k^{24\cdot(a_\mathcal{F}-1)})}$. Observe also that $\mathcal{E}_k(\mathcal{H})$ is a $K_{s_\mathcal{F}+k}$-minor-free graph class, and thus, by Proposition 4.2.1, we can always assume that G has $\mathcal{O}_{s_\mathcal{F}}(k\sqrt{\log k}\cdot n)$ edges, since otherwise we can directly conclude that (G,k) is a no-instance for the problem.

Run the algorithm `Find-Wall-Ed` from Lemma 8.3.3 with input (G, r_1, k) and, in time $2^{\mathcal{O}_{\ell_\mathcal{F}}(r_1^2+k^2)} \cdot n = 2^{2^{\mathcal{O}_{\ell_\mathcal{F}}(k^2\log k + c\log k)}} \cdot n$,

- either report a no-instance, or
- conclude that $\mathsf{tw}(G) \leq f_{8.3.3}(s_\mathcal{F}) \cdot r_1 + k$ and solve Elimination Distance to $\mathsf{exc}(\mathcal{F})$ in time $2^{\mathcal{O}_{\ell_\mathcal{F}}((r_1+k)k + (r_1+k)\log(r_1+k))} \cdot n = 2^{2^{\mathcal{O}_{\ell_\mathcal{F}}(k^2\log k + c\log k)}} \cdot n$ using the algorithm of Theorem 8.3.2, or
- obtain an r_1-wall W_1 of G.

If the output of Lemma 8.3.3 is a wall W_1, then run the algorithm `Clique-or-twFlat` of Proposition 4.6.3 with input $(G, r_2, s_{\mathcal{F}} + k)$. This takes time $2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2 \log k)} \cdot r_2^3 \log r_2} \cdot n = 2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2 \log k + c \log k)}} \cdot n$. If the result is a set A of size at most a and a regular flatness pair $(W_2, \mathfrak{R}_2)$ of $G - A$ of height r_2 whose $\mathfrak{R}_2$-compass has treewidth at most r_1, then proceed, otherwise output a no-answer.

Compute a W_2-canonical partition $\tilde{\mathcal{Q}}$ of $G - A$. Compute the set B of vertices of A that are adjacent to at least q p-internal bags of $\tilde{\mathcal{Q}}$. Apply the algorithm `Packing` of Proposition 4.6.8 with input $(l+1, r_3, p, G-A, \mathcal{R}_2, \tilde{\mathcal{Q}})$ to compute in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$ a collection $\mathcal{W} = \{W^1, ..., W^{l+1}\}$ of r_3-subwalls of W_2 respecting the properties of Proposition 4.6.8. By the choice of l, there is an $i \in [l+1]$ such that no vertex of $\bigcup \mathsf{influence}_{\mathfrak{R}_2}(W^i)$ is adjacent to a vertex of $A \setminus B$.

Run the algorithm from Proposition 4.6.6 with input $(G - B, W_2, \mathfrak{R}_2, W^i)$ to obtain a W^i-tilt $(W_3, \mathfrak{R}_3)$ of $(W_2, \mathfrak{R}_2)$ in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$.

After this, apply the algorithm `Homogeneous` of Proposition 4.6.12 with input $(r_4, a_{\mathcal{F}}-1, a, d, r_1, G, B, W_3, \mathfrak{R}_3)$, which, in time $2^{\mathcal{O}(c \cdot r_4 \log r_4 + r_1 \log r_1)} \cdot (n + m) = 2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2 \log k + c \log k)}} \cdot n$, outputs a tight flatness pair $(W_4, \mathfrak{R}_4)$ of $G - B$ of height r_4 that is d-homogeneous with respect to $\binom{B}{< a_{\mathcal{F}}}$ and is a W^*-tilt of $(W_3, \mathfrak{R}_3)$ for some subwall W^* of W_3.

Finally, apply the algorithm `Find-Irrelevant-Vertex` of Proposition 8.2.4 with input $(k(k+1)/2, a_{\mathcal{F}} - 1, G, B, W_4, \mathfrak{R}_4)$, which outputs, in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$, an irrelevant vertex v such that (G, k) and $(G - v, k)$ are equivalent instances of Elimination Distance to $\mathsf{exc}(\mathcal{F})$. Then the algorithm runs recursively on the equivalent instance $(G - v, k)$.

Since each run takes time $2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2 \log k + c \log k)}} \cdot n$ and there are at most n runs, the algorithm indeed runs in time $2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2 \log k + c \log k)}} \cdot n^2$.

Note that $c = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{24 \cdot (a_{\mathcal{F}} - 1)})}$, so if $\mathcal{F}$ contains an apex-graph, i.e., if $a_{\mathcal{F}} = 1$, then $c = \mathcal{O}_{\ell_{\mathcal{F}}}(1)$. Thus, the running time is $2^{2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{24 \cdot (a_{\mathcal{F}} - 1)})}}} \cdot n^2$ in the general case and $2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^2 \log k)}} \cdot n^2$ in the case where $\mathcal{F}$ contains an apex-graph.

8.3.3 Correctness of the algorithm

Let (G, k) be a yes-instance and let S be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. By running Lemma 8.3.3 with input (G, r_1, k), the algorithm should either get a report that $\mathsf{tw}(G) \leq f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + k$ or find an r_1-wall. The correctness of the former is obvious, so we will focus on the latter.

Let W_1 be an r_1-wall of G. According to Lemma 8.2.2, $K_{s_{\mathcal{F}}+k}$ is not a minor of G. Moreover, since W_1 is a wall of G of height r_1, $\mathsf{tw}(G) \geq \mathsf{tw}(W_1) \geq r_1 \geq f_{4.6.3}(s_{\mathcal{F}} + k) \cdot r_2$. Hence, if the algorithm runs `Clique-or-twFlat` of Proposition 4.6.3 with input $(G, r_2, s_{\mathcal{F}} + k)$, it should obtain a set A of size at most a and a regular flatness pair $(W_2, \mathfrak{R}_2)$ of $G - A$ of height r_2 whose $\mathfrak{R}_2$-compass has treewidth at most r_1.

As described in the algorithm, due to Proposition 4.6.8 and the fact that $r_2 \geq f_{4.6.8}(l + 1, r_3, p)$, there is an r_3-wall W^i that is a subwall of W_2 such that no vertex of $\bigcup \mathsf{influence}_{\mathfrak{R}_2}(W^i)$ is adjacent to a vertex of $A \setminus B$, where B is the set of vertices of A adjacent to at least q p-internal bags of a W_2-canonical partition $\tilde{\mathcal{Q}}$ of $G - A$.

When the algorithm applies Proposition 4.6.6 with input $(G-B, W_2, \mathfrak{R}_2, W^i)$, it obtains a W^i-tilt $(W_3, \mathfrak{R}_3)$ of $(W_2, \mathfrak{R}_2)$. Due to Observation 4.6.4 and Observation 4.6.5, $(W_3, \mathfrak{R}_3)$ is a regular flatness pair of $G-B$ whose $\mathfrak{R}_5$-compass has treewidth at most r_1. Thus, since $r_3 = f_{4.6.12}(r_4, a_{\mathcal{F}}-1, a, d)$, the algorithm can apply `Homogeneous` of Proposition 4.6.12 with input $(r_4, a_{\mathcal{F}} - 1, a, d, r_1, G, B, W_3, \mathfrak{R}_3)$

to obtain a tight flatness pair $(W_4, \mathfrak{R}_4)$ of $G - B$ of height r_4 that is d-homogeneous with respect to $\binom{B}{<a_{\mathcal{F}}}$ and is a W^*-tilt of $(W_3, \mathfrak{R}_3)$ for some subwall W^* of W_3. According to Observation 4.6.5, $(W_4, \mathfrak{R}_4)$ is regular.

Let S' be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. Lemma 8.2.7 implies that there is a set $X_{S'} \supseteq S'$ such that $G - X_{S'} \in \mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-A,W_2}(X_{S'}) \leq k(k+1)/2$.

Since $r_2 \geq f_{8.2.5}(a_{\mathcal{F}}, s_{\mathcal{F}}, k(k+1)/2)$, every subset of B of size $a_{\mathcal{F}}$ intersects $X_{S'}$ according to Proposition 8.2.5. Hence, $|B \setminus X_{S'}| \leq a_{\mathcal{F}} - 1$.

Moreover, note that $(W_3, \mathfrak{R}_3)$ is a W^i-tilt of $(W_2, \mathfrak{R}_2)$, $(W_4, \mathfrak{R}_4)$ is a W^*-tilt of $(W_3, \mathfrak{R}_3)$, and $(W_4, \mathfrak{R}_4)$ is a flatness pair of $G - B$ with $B \subseteq A$. Thus, given a W_4-canonical partition Q_1 of $G - B$, there is a W_2-canonical partition Q_2 of $G - A$ such that each internal bag of Q_1 is an internal bag of Q_2. Thus, $\mathsf{bid}_{G-B,W_4}(X_{S'}) \leq \mathsf{bid}_{G-A,W_2}(X_{S'})$.

Hence, the algorithm can apply `Find-Irrelevant-Vertex` of Proposition 8.2.4 with input $(k(k+1)/2, a_{\mathcal{F}} - 1, G, B, W_4, \mathfrak{R}_4)$ and obtain a vertex v such that, for any k-elimination set S' of G for $\mathsf{exc}(\mathcal{F})$, $G - X_{S'} \in \mathsf{exc}(\mathcal{F})$ if and only if $G - (X_{S'} \setminus v) \in \mathsf{exc}(\mathcal{F})$. Thus, there is a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$ if and only if there is a k-elimination set of $G - v$ for $\mathsf{exc}(\mathcal{F})$. It follows that (G, k) and $(G - v, k)$ are equivalent instances of Elimination Distance to $\mathsf{exc}(\mathcal{F})$.

Suppose now that (G, k) is a no-instance. Note that as long as Proposition 4.6.3 outputs a flatness pair $(W_2, \mathfrak{R}_2)$, what follows in the proof of correctness works even if (G, k) is a no-instance. Therefore, we will find an irrelevant vertex. Otherwise, we would have declared a no-instance beforehand. Thus, Theorem 8.3.1 follows.

8.4 Elimination distance when excluding an apex-graph

In the case where $\mathcal{F}$ contains an apex-graph, we obtain an alternative algorithm whose complexity is single-exponential in k and cubic in n. The following theorem is a restatement of Theorem 2.4.2.

Theorem 8.4.1. *For every finite collection of graphs $\mathcal{F}$ that contains an apex-graph, there exists an algorithm that, given a graph G and an integer k, decides whether $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \leq k$ in time $2^{k^{\mathcal{O}_{\ell_{\mathcal{F}}}(1)}} \cdot n^3$.*

We are now in Option 2 of Section 8.1. That is, contrary to the previous section, since $a_{\mathcal{F}} = 1$, any vertex fulfilling the criteria of Proposition 8.2.5 belongs to every k-elimination set S of the input graph for $\mathsf{exc}(\mathcal{F})$. Hence, we can add a step (Step 3 of Subsection 8.4.2) very similar to the Step 5 of Subsection 7.3.2. Here, a k-elimination set may have size $\Omega(n)$, so we may run Step 3 $\Omega(n)$ times. Since our Step 3 below runs in quadratic time, this gives the cubic dependence of this algorithm. Fortunately, since we apply this Step 3, contrary to Section 8.3, we manage to find a flatness pair along with an apex set whose size does not depend on k. Hence, when applying Proposition 4.6.12, we do not get a triple-exponential dependence on k anymore for the size of the wall we need to find originally.

In order to remember the vertices that are found to belong to every k-elimination set, since they do not decrease k, we need to distinguish them in the input. Hence, we actually give here an algorithm to solve a more general problem with annotations described in Subsection 8.4.1.

8.4.1 Generalization to annotated elimination distance

Contrary to the previous section, since $a_{\mathcal{F}} = 1$, when applying Proposition 8.2.5, we find a vertex that belongs to every k-elimination set S. Such vertices are taken into account by considering the following generalization of Elimination Distance to $\mathsf{exc}(\mathcal{F})$.

Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$

Input:	A graph G, a set $S_0 \subseteq V(G)$, and a non-negative integer k.
Task:	Find, if it exists, a k-elimination set S of G for the class $\mathsf{exc}(\mathcal{F})$ such that $S_0 \subseteq S$.

S_0 is a set of annotated vertices that corresponds to the vertices identified as vertices of every k-elimination set S while running the algorithm. Clearly, Elimination Distance to $\mathsf{exc}(\mathcal{F})$ is the particular case of Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ when S_0 is empty.

In the following lemma, we generalize Theorem 8.3.2 to its "annotated" version. More precisely, we present a simple trick to reduce the above problem to its "unannotated" version while not changing the treewidth of the input graph so much.

Lemma 8.4.2. *Let $\mathcal{F}$ be a finite collection of graphs. There is an algorithm that, given a graph G, a set $S_0 \subseteq V(G)$, and two integers k and tw such that the treewidth of G is bounded by tw, decides whether (G, S_0, k) is a yes-instance of* Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ *in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{tw}\cdot k+\mathsf{tw}\log\mathsf{tw})}\cdot n$.*

Proof. Given a minor-closed graph class $\mathcal{H}$, let $\mathcal{C}(\mathcal{H}) := \{G \mid \forall C \in \mathsf{cc}(G), C \in \mathcal{H}\}$. Bulian and Dawar [44] showed that if $H \in \mathsf{obs}(\mathcal{H})$ has l connected components, then each graph $\bar{H}$ obtained from H by adding $l-1$ edges to obtain a connected graph belongs to $\mathsf{obs}(\mathcal{C}(\mathcal{H}))$. Thus, let $H_{\mathcal{F}} \in \mathsf{obs}(\mathcal{C}(\mathsf{exc}(\mathcal{F}))))$ obtained in such a way. As said above, $H_{\mathcal{F}}$ is connected.

Let G be a graph of treewidth at most tw and S_0 be a subset of $V(G)$. Let G' be a graph obtained from G by gluing a graph H_v isomorphic to $H_{\mathcal{F}}$ to each vertex v of S_0, where v is identified with an arbitrarily chosen vertex of H_v.

Let us show that (G, S_0, k) is a yes-instance of Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ if and only if (G', k) is a yes-instance of Elimination Distance to $\mathcal{C}(\mathsf{exc}(\mathcal{F}))$. If $S_0 = \emptyset$, the proof is trivial, so we suppose $S_0 \neq \emptyset$.

Let (F', χ', R') be a $\mathcal{C}(\mathsf{exc}(\mathcal{F})))$-elimination forest of G' of height at most k with associated k-elimination set S'. For each $v \in S_0$, the fact that $H_v \notin \mathcal{C}(\mathsf{exc}(\mathcal{F})))$ implies that $V(H_v) \cap S' \neq \emptyset$. Let $y \in V(F')$ be the least common ancestor of $\chi'^{-1}(H_v)$ in F'. Since H_v is connected, according to Lemma 8.2.1, y exists and belongs to $\chi'^{-1}(H_v)$. Moreover, since $V(H_v) \cap S' \neq \emptyset$, $y \in \mathsf{Int}(F', R')$.

Let (F'', χ'', R'') be the $\mathcal{F}$-elimination forest of G obtained from (F', χ', R') as follows. For every $v \in S_0$ and every $t \in \chi'^{-1}(H_v - v)$ if $t \in \mathsf{Int}(F', R')$, remove t from F'' and add edges between the parent and the children of t and if $t \in \mathsf{Leaf}(F', R')$, remove $H_v - v$ from $\chi''(t)$. If $G[\chi''(t)]$ is not connected, then we update F'' by replacing t by $|\mathsf{cc}(G[\chi''(t)])|$ nodes, each one associated with a connected component of $G[\chi''(t)]$.

Thus, we have an $\mathcal{F}$-elimination forest of G associated with a k-elimination set S with $S_0 \subseteq S$ and with height at most k, implying that (G, S_0, k) is a yes-instance of Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$.

Conversely, given an $\mathcal{F}$-elimination forest (F, χ, R) of G of height at most k, and $S_0 \subseteq S := \chi(\mathsf{Int}(F, R))$, we can obtain a $\mathsf{obs}(\mathcal{C}\mathsf{exc}(\mathcal{F})))$-elimination forest (F', χ', R') of G' of height at most k by adding to each node $\chi^{-1}(v)$ for $v \in S_0 \subseteq S$ a leaf associated with $H_v - v \in \mathcal{C}(\mathsf{exc}(\mathcal{F})))$. The height of (F', χ', R') is indeed still at most k. Therefore, (G, S_0, k) is a yes-instance for Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ if and only if (G', k) is a yes-instance for Elimination Distance to $\mathcal{C}(\mathsf{exc}(\mathcal{F}))$.

Thus, if we apply the algorithm of Theorem 8.3.2 to (G', k) to solve Elimination Distance to $\mathcal{C}(\mathsf{exc}(\mathcal{F}))$, we solve Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ on instance (G, S_0, k) in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{tw}(G')\cdot k+\mathsf{tw}(G')\log\mathsf{tw}(G')))}\cdot |V(G')|$.

Given a tree decomposition $\mathcal{T} = (T, \beta)$ of G of width t, we can obtain a tree decomposition $\mathcal{T}'$ of G' of width at most $t + |V(H_{\mathcal{F}})|$ by adding a node x_v for each $v \in S_0$ such that $\beta(v) = V(H_v)$, adjacent to a node y of T such that $v \in \beta(y)$. Thus, $\mathsf{tw}(G') \leq \mathsf{tw}(G) + |V(H_{\mathcal{F}})| = \mathsf{tw}(G) + \mathcal{O}_{\ell_{\mathcal{F}}}(1)$. Moreover, $|V(G')| = |V(G)| + (|V(H_{\mathcal{F}})| - 1) \cdot |S_0| = \mathcal{O}_{\ell_{\mathcal{F}}}(|V(G)|)$. Therefore, we can solve Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ on instance (G, S_0, k) in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{tw} \cdot k + \mathsf{tw} \log \mathsf{tw}))} \cdot n$, and the lemma follows. □

8.4.2 Description of the algorithm for Elimination Distance to $\mathsf{exc}(\mathcal{F})$ when $a_{\mathcal{F}} = 1$

We now describe the algorithm to solve Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$, and hence Elimination Distance to $\mathsf{exc}(\mathcal{F})$, when $a_{\mathcal{F}} = 1$, i.e., when $\mathcal{F}$ contains an apex-graph. Note that, similarly to this algorithm, the one from Section 8.3 in the general case can also very easily be generalized to its "annotated" version. We stress that the reason for the better parametric dependence of this algorithm compared to the algorithm of Theorem 8.3.1 is that we pursue homogeneous flat walls where homogeneity is asked for subsets of size *not* depending on k.

We define the following constants.

$$\begin{aligned}
a &= g_{4.6.2}(s_{\mathcal{F}}), & q &= g_{8.2.5}(1, s_{\mathcal{F}}, k(k+1)/2),\\
p &= h_{8.2.5}(1, s_{\mathcal{F}}, k(k+1)/2), & l &= (q-1) \cdot (k+a),\\
d &= g_{8.2.4}(a, \ell_{\mathcal{F}}) & r_4 &= f_{8.2.4}(a, \ell_{\mathcal{F}}, 3, k(k+1)/2),\\
r_3 &= f_{4.6.12}(r_4, a, a, d), & t &= f_{4.6.3}(s_{\mathcal{F}}) \cdot r_3,\\
r_2 &= \mathsf{odd}(t+3), & r_2' &= \mathsf{odd}(\max\{f_{8.2.5}(1, s_{\mathcal{F}}, k(k+1)/2), f_{4.6.8}(l+1, r_2, p)\}),\\
r_1' &= \mathsf{odd}(f_{4.6.2}(s_{\mathcal{F}}) \cdot r_2'), & r_1 &= \mathsf{odd}(r_1' + k).
\end{aligned}$$

Note that $r_4 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^2)$, $r_3, r_2 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c})$, and $r_2', r_1', r_1 = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c+7/2})$, where $c = g_{4.6.12}(a, a, d) = \mathcal{O}_{\ell_{\mathcal{F}}}(1)$. Recall that we assume that G has $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$ edges.

The input of this algorithm is a graph G, a set $S_0 \subseteq V(G)$, and an integer k.

Step 1. Run the algorithm `Find-Wall-Ed` from Lemma 8.3.3 with input $(G - S_0, r_1, k)$ and, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r_1^2 + k^2)} \cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{4c+7})} \cdot n$,

- either report a no-instance, or
- conclude that $\mathsf{tw}(G - S_0) \leq f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + k$ and solve Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$ with input $(G, S_0, f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + 2k, k)$ in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}((r_1+k) \cdot k + (r_1+k)\log(r_1+k)))} \cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c+9/2})} \cdot n$ using the algorithm of Lemma 8.4.2, or
- obtain an r_1-wall W_1 of G.

If the output of Lemma 8.3.3 is a wall W_1, consider all the $\binom{r_1}{r_2}^2 = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c} \log k)}$ r_2-subwalls of W_1 and for each one of them, say W_2, let W_2^* be the central $(r_2 - 2)$-subwall of W_2 and let D_{W_2} be the graph obtained from $G - S_0$ after removing the perimeter of W_2 and taking the connected component containing W_2^*. Run the algorithm `Clique-or-twFlat` of Proposition 4.6.3 with input $(D_{W_2}, r_3, s_{\mathcal{F}})$. This takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r_3^2)} \cdot n = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{4c})} \cdot n$. If for one of these subwalls the result is a set A of size at most a and a regular flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} - A$ of height r_3 whose $\mathfrak{R}_3$-compass has treewidth at most t, then we proceed to Step 2, otherwise proceed to Step 3.

Step 2. We obtain a 5-tuple $\mathfrak{R}_3'$ by adding all vertices of $G - (S_0 \cup V(\mathsf{Compass}_{\mathfrak{R}_3}(W_3)))$ to the set in the first coordinate of $\mathfrak{R}_3$, such that $(W_3, \mathfrak{R}_3')$ is a regular flatness pair of $G - (S_0 \cup A)$.

We apply the algorithm `Homogeneous` of Proposition 4.6.12 with input $(r_4, a, a, d, t, G-S_0, A, W_3, \mathfrak{R}_3')$, which outputs, in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(r_4 \log r_4 + t \log t)} \cdot (n+m) = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c} \log k)} \cdot n$ a tight flatness pair $(W_4, \mathfrak{R}_4)$ of $G - (S_0 \cup A)$ of height r_4 that is d-homogeneous with respect to 2^A and is a W^*-tilt of $(W_3, \mathfrak{R}_3')$ for some subwall W' of W. At this point, we stress that the reason for the better parametric dependence of this algorithm compared to the previous one comes from the fact that the third input parameter a in `Homogeneous` does *not* depend of k. We apply the algorithm `Find-Irrelevant-Vertex` of Proposition 8.2.4 with input $(k(k+1)/2, a, G - S_0, A, W_4, \mathfrak{R}_4)$, which outputs, in time $\mathcal{O}_{s_{\mathcal{F}}}(k\sqrt{\log k} \cdot n)$, a vertex v such that (G, S_0, k) and $(G - v, S_0, k)$ are equivalent instances of Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$. Then the algorithm runs recursively on the equivalent instance $(G - v, S_0, k)$.

Step 3. Consider all the r_2'-subwalls of W_1, which are $\binom{r_1}{r_2'}^2 = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c+7/2} \log k)}$ many, and for each of them, say W_2', compute its canonical partition $\mathcal{Q}$. Then, contract each bag Q of $\mathcal{Q}$ to a single vertex v_Q, remove the vertices v_Q where Q is not a p-internal bag of $\mathcal{Q}$, and add a new vertex v_{all} and make it adjacent to all remaining v_Q's. In the resulting graph G', for every vertex y of $G - S_0 - V(W_2')$, check, in time $\mathcal{O}(q \cdot m) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^7\sqrt{\log k} \cdot n)$, using a flow augmentation algorithm [87], whether there are q internally vertex-disjoint paths from v_{all} to y. Let $\tilde{A}$ be the set of such y's.

If $\tilde{A} = \emptyset$, then report a no-instance.

If $1 \leq |\tilde{A}| \leq k + a$, then each vertex of $\tilde{A}$ should intersect every k-elimination set S of G for $\mathsf{exc}(\mathcal{F})$. The algorithm runs recursively on $(G, S_0 \cup \tilde{A}, k)$.

If, for every wall, $|\tilde{A}| > k + a$, then report that (G, S_0, k) is a no-instance of Annotated Elimination Distance to $\mathsf{exc}(\mathcal{F})$.

After Step 2, the size of G decreases by one, so Step 2 can be applied at most n times. After Step 3, the size of S_0 increases by at least one, so Step 3 can also be applied at most n times. Note that, if $S_0 = V(G)$, then $\mathsf{tw}(G - S_0) = 0$, so the algorithm stops. Thus, the algorithm finishes. Notice also that Step 3, when applied, takes time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c+7/2} \log k)} \cdot n^2$, because we apply a flow algorithm for each of the $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c+7/2} \log k)}$ r_2'-subwalls and for each vertex of G. Since Step 1 and Step 2 run in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{4c+7})} \cdot n$ and $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{2c} \log k)} \cdot n$, respectively, and both may be applied at most n times, the claimed time complexity follows: the algorithm runs in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(k^{4c+7})} \cdot n^3$.

8.4.3 Correctness of the algorithm

Let (G, S_0, k) be a yes-instance and let S be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$ with $S_0 \subseteq S$. By running Lemma 8.3.3 with input $(G - S_0, r_1, k)$, the algorithm should either get a report that $\mathsf{tw}(G - S_0) \leq f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + k$ or find an r_1-wall.

If $\mathsf{tw}(G - S_0) \leq f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + k$, then since $S_0 \subseteq S$, $\mathsf{tw}(G - S) \leq f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + k$. Hence, according to Lemma 8.2.3, $\mathsf{tw}(G) \leq f_{8.3.3}(s_{\mathcal{F}}) \cdot r_1 + 2k$.

Otherwise, let W_1 be an r_1-wall of $G - S_0$. According to Lemma 8.2.6, since $r_1 \geq r_1' + k$, there is an r_1'-subwall W_1' of W_1 that is a subwall of $G - S$. Let H be the connected component of $G - S$ containing W_2. The fact that H belongs to $\mathsf{exc}(\mathcal{F})$ implies that it has no $K_{s_{\mathcal{F}}}$-minor. Therefore, by Proposition 4.6.2, since $r_1' \geq f_{4.6.2}(s_{\mathcal{F}}) \cdot r_2'$, there is a set $B \subseteq V(H)$, with $|B| \leq a$, and a flatness pair $(W_2', \mathfrak{R}_2')$ of $H - B$ of height r_2'.

Let $\mathcal{Q}$ be the canonical partition of W_2'. Let G' be the graph obtained after contracting every bag Q of $\mathcal{Q}$ to a single vertex v_Q, removing the vertices v_Q where Q is not a p-internal bag of $\mathcal{Q}$, and adding a new vertex v_{all} and making it adjacent to all remaining v_Q's. Let $\tilde{A}$ be the set of vertices y of $G - V(W_2')$ such that there are q internally vertex-disjoint paths from v_{all} to y in G'. Since S is a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$, there is a set $P \subseteq S$ of size at most k so that $(L, R) := (V(G) \setminus V(H), V(H) \cup P)$ is a separation of G with $P = L \cap R$.

Note that $\tilde{A} \subseteq P \cup B$. To show this, we first prove that, for every $y \notin P \cup B$, the maximum number of internally vertex-disjoint paths from v_{all} to y in G' is less than q. Indeed, if y is a vertex in $(V(G) \setminus V(H)) \setminus P$, then every path from y to a vertex of W_2' intersects P. Therefore, there are at most $k < q$ internally vertex-disjoint paths from v_{all} to such a $y \in (V(G) \setminus V(H)) \setminus P$ in G'. If $y \in V(H) \setminus B$, then we distinguish two cases. First, if y is a vertex in the $\mathfrak{R}_2'$-compass of W_2', there are at most $k + a$ such paths that intersect the set $P \cup B$ and at most four paths that do not intersect $P \cup B$ (in the graph $G' - (P \cup B)$) due to the flatness of W_2'. If y is in $V(H)$ but not a vertex in the $\mathfrak{R}_2'$-compass of W_2', then, since by the definition of flatness pairs the perimeter of W_2' together with the set $P \cup B$ separate y from the $\mathfrak{R}_2'$-compass of W_2', every collection of internally vertex-disjoint paths from v_{all} to y in G' should intersect the set $\{v_{Q_{\text{ext}}}\} \cup P \cup B$, where Q_{ext} is the external bag of $\mathcal{Q}$. Therefore, in all cases, if $y \notin P \cup B$, the maximum number of internally vertex-disjoint paths from v_{all} to y in G' is at most $k + a + 4 < q$. Therefore, $y \notin \tilde{A}$. Hence, $|\tilde{A}| \leq k + a$.

Let $\mathfrak{R}_2''$ be the 5-tuple obtained by adding all vertices of $G - S_0 - P - H$ to the set in the first coordinate of $\mathfrak{R}_2'$. Notice that since every path between $G - H$ and H intersects P, $(W_2', \mathfrak{R}_2'')$ is a flatness pair of $G - (P \cup B)$.

If $\tilde{A} = \emptyset$, then let $\tilde{\mathcal{Q}}$ be an enhancement of $\mathcal{Q}$ on $G - (P \cup B)$. No vertex of $(P \cup B) \setminus S_0$ is adjacent to vertices of at least q p-internal bags of $\tilde{\mathcal{Q}}$. This means that the p-internal bags of $\tilde{\mathcal{Q}}$ that contain vertices adjacent to some vertex of $P \cup B$ are at most $(q - 1) \cdot (k + a) = l$.

Consider a family $\mathcal{W} = \{W^1, \dots, W^{l+1}\}$ of $l + 1$ r_2-subwalls of W_2' such that for every $i \in [l + 1]$, $\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^i)$ is a subgraph of $\bigcup\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{\mathcal{Q}}\}$ and for every $i, j \in [l + 1]$, with $i \neq j$, there is no internal bag of $\tilde{\mathcal{Q}}$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^j))$. The existence of $\mathcal{W}$ follows from Proposition 4.6.8 and the fact that $r_2' \geq f_{4.6.8}(l + 1, r_2, p)$.

The fact that the p-internal bags of $\tilde{\mathcal{Q}}$ that contain vertices adjacent to some vertex of $(P \cup B) \setminus S_0$ are at most l implies that there exists an $i \in [l + 1]$ such that no vertex of $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^i))$ is adjacent, in G, to a vertex in $(P \cup B) \setminus S_0$.

Let $W_2 := W^i$, let W_2^* be the central $(r_2 - 2)$-subwall of W_2, and let D_{W_2} be the graph obtained from $G - S_0$ after removing the perimeter of W_2 and taking the connected component containing W_2^*. Any path going from a vertex in H to a vertex in $G - H$ intersects P. Thus, $D_{W_2} \subseteq H$ and therefore, $K_{s_{\mathcal{F}}}$ is not a minor of D_{W_2}. Moreover, W_2^* is a wall of D_{W_2} of height $r_2 - 2 \geq t + 1$, so $\mathsf{tw}(D_{W_2}) > t = f_{4.6.3}(s_{\mathcal{F}}) \cdot r_3$. Therefore, if the algorithm runs `Clique-or-twFlat` of Proposition 4.6.3 with input $(D_{W_2}, r_3, s_{\mathcal{F}})$, it should obtain a set A of size at most a and a regular flatness pair $(W_3, \mathfrak{R}_3)$ of $D_{W_2} - A$ of height r_3 whose $\mathfrak{R}_3$-compass has treewidth at most t. Hence, the algorithm then runs Step 2.

If $\tilde{A} \neq \emptyset$, then recall that for every $y \in \tilde{A}$, y has q internally vertex-disjoint paths $P_1, \dots, P_q$ to different p-internal bags $Q_1, \dots, Q_q$ of $\mathcal{Q}$ in G. Hence, there is an enhancement $\tilde{\mathcal{Q}}_y$ of $\mathcal{Q}$ on $G - (P \cup B)$ such that P_i belongs to the bag $\tilde{Q}_i$ that extends Q_i for $i \in [q]$. Therefore, y is adjacent to vertices of at least q p-internal bags of $\tilde{\mathcal{Q}}_y$. Let S' be a k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. According to Lemma 8.2.7, there is a set $X_{S'} \subseteq V(G)$ such that $G - X_{S'} \in \mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-(P \cup B), W_2'}(X_{S'}) \leq k(k + 1)/2$. Therefore, $y \in X_{S'}$ due to Proposition 8.2.5 and the fact that

$r_2' \geq f_{8.2.5}(1, s_{\mathcal{F}}, k(k+1)/2)$. Let $C_{S'} := G - X_{S'}$. Recall that y is adjacent to $q > k(k+1)/2$ p-internal bags of $\tilde{\mathcal{Q}}_y$. However, $\mathsf{bid}_{\tilde{\mathcal{Q}}_y}(X_{S'}) \leq \mathsf{bid}_{G-(P\cup B),W_2'}(X_{S'}) \leq k(k+1)/2$. Therefore, y is adjacent to $C_{S'}$, so $y \in S'$. Since for every $y \in \tilde{A}$, for every k-elimination set S', we have $y \in S'$, it implies that $\tilde{A}$ is included in every k-elimination set of G for $\mathsf{exc}(\mathcal{F})$. Hence, if the algorithm runs Step 3, it then recursively runs on the equivalent instance $(G, S_0 \cup \tilde{A}, k)$.

We do not suppose that (G, k) is a yes-instance anymore. Let us show the correctness of Step 2. Suppose that we obtained the wanted flatness pair $(W_3, \mathfrak{R}_3)$ in Step 1. We obtain a 5-tuple $\mathfrak{R}_3'$ by adding all vertices of $G - (S_0 \cup V(\mathsf{Compass}_{\mathfrak{R}_3}(W_3)))$ to the set in the first coordinate of $\mathfrak{R}_3$. Since $(W_3, \mathfrak{R}_3)$ is a regular flatness pair of $D_{W^i} - A$ whose $\mathfrak{R}_3$-compass has treewidth at most t and since the vertices added in $\mathfrak{R}_3'$ are only adjacent to the perimeter of W^i, it follows that $(W_3, \mathfrak{R}_3')$ is a regular flatness pair of $G - (S_0 \cup A)$ whose $\mathfrak{R}_3'$-compass has treewidth at most t.

If the algorithm applies the algorithm `Homogeneous` of Proposition 4.6.12 with $(r_4, a, a, d, t, G - S_0, A, W_3, \mathfrak{R}_3')$ as input, it obtains a tight flatness pair $(W_4, \mathfrak{R}_4)$ of $G - (S_0 \cup A)$ of height r_4 that is d-homogeneous with respect to 2^A and is a W^*-tilt of $(W_3, \mathfrak{R}_3')$ for some subwall W' of W. According to Observation 4.6.5, $(W_4, \mathfrak{R}_4)$ is regular.

Lemma 8.2.7 implies that for every k-elimination set $S' \supseteq S_0$, there is a set $X_{S'} \supseteq S'$ with $\mathsf{bid}_{G-(S_0\cup A),W_4}(X_{S'}) \leq k(k+1)/2$ and $G - X_{S'} \in \mathsf{exc}(\mathcal{F})$. We have that $|A \setminus X| \leq |A| \leq a$, so the algorithm can apply `Find-Irrelevant-Vertex` of Proposition 8.2.4 with input $(k(k+1)/2, a, G - S_0, A, W_4, \mathfrak{R}_4)$ to obtain a vertex v such that for every k-elimination set $S' \supseteq S_0$, $G - X_{S'} \in \mathsf{exc}(\mathcal{F})$ if and only if $G - (X_{S'} \setminus v) \in \mathsf{exc}(\mathcal{F})$. It follows that (G, S_0, k) and $(G - v, S_0, k)$ are equivalent instances of ANNOTATED ELIMINATION DISTANCE TO $\mathsf{exc}(\mathcal{F})$.

Suppose now that (G, S_0, k) is a no-instance. In Step 1, the algorithm either reports a no-instance or finds a wall. In the latter case, the algorithm either goes to Step 2 or to Step 3. If it runs Step 2, the previous paragraph justifies that the algorithm finds a vertex v such that $(G - v, S_0, k)$ is a no-instance. If the algorithm runs Step 3, then it either reports a no-instance or recursively runs on the instance $(G - y, S_0 \cup \tilde{A}, k)$. If $(G - y, S_0 \cup \tilde{A}, k)$ is yes-instance, then so is (G, k). Thus, $(G - y, S_0 \cup \tilde{A}, k)$. is a no-instance. Hence, the algorithm always report a no-instance. Therefore, Theorem 8.4.1 follows.

Constructing the elimination ordering. Notice that the results of Theorem 8.3.1 and Theorem 8.4.1 solve the decision version of ELIMINATION DISTANCE TO $\mathcal{H}$. Using the dynamic programming algorithm of Section 8.5, we may find a k-elimination set X certifying that $\mathsf{ed}_{\mathcal{H}}(G) \leq k$. One may further determine, from X, the way the elimination ordering is applied on the vertices of X as follows. Let $\mathsf{torso}(G, X)$ be the graph obtained from $G[X]$ if, for every connected component C of $G - X$, we make adjacent all pairs of vertices in $N_G(V(C))$ in $G[X]$. Then we know that $\mathsf{td}(\mathsf{torso}(G, X)) \leq k$ and the required elimination ordering is the same as the one for $\mathsf{torso}(G, X)$, which can be computed by the algorithm of [256] in time $2^{\mathcal{O}(k^2)} \cdot n$.

8.5 Solving ELIMINATION DISTANCE TO $\mathsf{exc}(\mathcal{F})$ on tree decompositions

In this section, we prove Theorem 8.3.2.

Bodlaender, Gilbert, Kloks, and Hafsteinsson give a relation between the treedepth and the treewidth of a graph in [36].

Proposition 8.5.1 ([36]). *Let G be a graph with n vertices. Then* $\mathsf{tw}(G) \leq \mathsf{td}(G) \leq \mathsf{tw}(G) \cdot \log n$.

Since $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \le \mathsf{td}(G) \le \mathsf{tw}(G) \cdot \log n$ and $\mathsf{td}(G) \le \mathsf{tw}(G) \cdot \log n$, Theorem 8.3.2 implies the existence of an XP-algorithm for Elimination Distance to $\mathsf{exc}(\mathcal{F})$ parameterized by treewidth.

Corollary 8.5.2. *For every finite collection of graphs $\mathcal{F}$, there exists an algorithm that, given a graph G of treewidth at most* tw*, computes $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G)$ in time $n^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{tw}^2)}$.*

According to Proposition 8.5.1 again, $\mathsf{tw}(G) \le \mathsf{td}(G)$ for any graph G. Since we moreover have $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \le \mathsf{td}(G)$, Theorem 8.3.2 implies the existence of an FPT-algorithm for Elimination Distance to $\mathsf{exc}(\mathcal{F})$ parameterized by treedepth.

Corollary 8.5.3. *For every finite collection of graphs $\mathcal{F}$, there exists an algorithm that, given a graph G of treedepth at most* td*, computes $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G)$ in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\mathsf{td}^2)} \cdot n$.*

Our algorithm takes inspiration from the dynamic programming algorithm of Reidl, Rossmanith, Villaamil, and Sikdar [256] for treedepth.

Proposition 8.5.4 ([256])**.** *Given a graph G, a tree decomposition of G of width w, and an integer k, there is an algorithm that decides whether $\mathsf{td}(G) \le k$ in time $2^{\mathcal{O}(k \cdot w)} \cdot n$.*

If $\mathcal{F} = \{K_1\}$, elimination distance reduces to treedepth. Recall that Elimination Distance to $\{K_1\}$ is the problem asking whether $\mathsf{td}(G) \le k$, which admits an algorithm in time $2^{\mathcal{O}(k \cdot w)} \cdot n$ because of Proposition 8.5.4. Therefore, we may assume throughout this section that $\mathcal{F}$ is non-trivial in order to have the useful property that a graph with a single vertex belongs to $\mathcal{H} = \mathsf{exc}(\mathcal{F})$. This will simplify the algorithm. Moreover, the elimination distance to $\mathsf{exc}(\mathcal{F})$ of a disconnected graph is the maximum of the elimination distance of its connected components, and therefore, we may assume that the considered graphs and boundaried graphs are connected.

Just as in Section 7.6, we use the representative-based framework introduced in [24], that we combine here with ideas from [256], to create our dynamic programming algorithm.

In order to describe our dynamic programming algorithm, we have to describe its corresponding tables, encode "partial elimination sets", and show how to calculate this information using a nice tree decomposition of the input graph. For this reason, in Subsection 8.5.1 we define annotated trees. Annotated trees are labeled rooted trees that come together with a boundaried graph such that the annotated nodes of the tree are mapped to the vertices of the boundaried graph with the same label. This notion is used in Subsection 8.5.2 in order to define the *characteristic* of a boundaried graph, which intuitively encodes how partial elimination trees can be present inside the boundaried graph. Forget, introduce, and join procedures that shall be used in the dynamic program on nice tree decompositions are presented in Subsection 8.5.3. In Subsection 8.5.4, we present the dynamic program and prove its correctness. We conclude this section with Subsection 8.5.5, where we show that boundaried graphs with the same characteristic can be exchanged, i.e., give graphs of the same elimination distance to $\mathcal{F}$ when "glued" to the same boundaried graph. This latter result will also be used in Section 8.6.

Some additional notations. We denote by $\mathsf{Im}(f)$ the image of a function f and by $\mathsf{Ker}(f)$ its kernel, i.e., the elements whose image by f is 0. $(i \leftrightarrow j)$ denotes the transposition of i and j, for i, j in some set $I \subseteq \mathbb{N}$.

For every $q \in V(T)$, we set $G_q^{(T,\beta,r)} = G[\beta(T_q)]$. We may write G_q instead of $G_q^{(T,\beta,r)}$ when there is no ambiguity about (T, β, r).

8.5.1 Annotated trees

We proceed to define annotated trees, which we will use to codify the tables of our dynamic program.

Annotated trees. An *annotated tree* is a tuple $\hat{T} = (T, r, h, \mathbf{R}, f)$, where (T, r) is a rooted tree, $h : V(T) \to \mathbb{N}$, $\mathbf{R} = (R, B, \phi)$ is a boundaried graph, and $f : [|B|] \to V(T)$. See Figure 8.1 for an illustration of an annotated tree. We stress that different integers in $[|B|]$ can be mapped, via f, to the same node of T. The *trivial annotated tree*, denoted by $\hat{\mathbb{1}}$, is $(T, r, h, \mathbb{1}, f)$ where T is the rooted tree with a single node r, h is the constant function 0, $\mathbb{1}$ is the boundaried graph with one single vertex that is also part of the boundary, and f maps 1 to r. The *height* of an annotated tree is $h(r)$. Given an annotated tree $\hat{T} = (T, r, h, (R, B, \phi), f)$, we refer to (T, r) as its *rooted tree*. Given an annotated tree $\hat{T} = (T, r, h, (R, B, \phi), f)$ and a permutation σ of $[|B|]$, we use $\sigma(\hat{T})$ to denote $(T, r, h, (R, B, \sigma \circ \phi), f \circ \sigma^{-1})$.

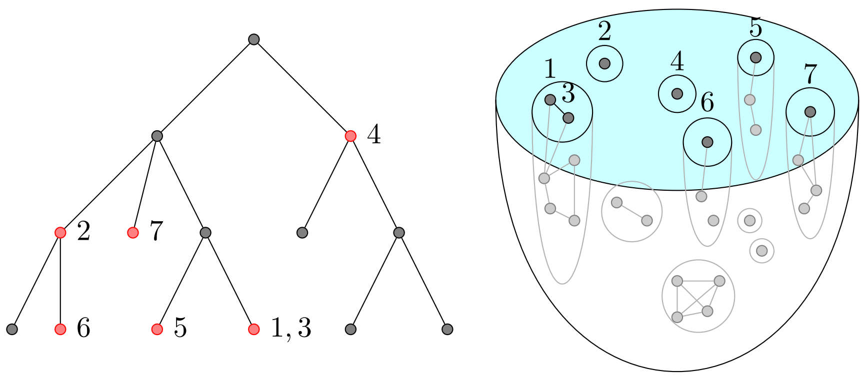

Figure 8.1: An annotated tree made of a rooted tree (left) and a boundaried graph (right). The numbers in the left figure correspond to the pre-images of f for the nodes of $V(T)$ and the numbers on the right figure correspond to the images of ϕ. The function h that gives a value to each node of the tree is not represented.

We also define the following operations on annotated trees, which will be used to combine the tables of the dynamic programming algorithm. The first one is inspired by a similar operation introduced in [256].

Crop operation. Given an annotated tree $\hat{T} = (T, r, h, (R, B, \phi), f)$, the *crop operation*, denoted by $\mathsf{crop}(\hat{T})$, outputs the annotated tree obtained from $\hat{T}$ by iteratively removing the leaves of T that are not in $\mathsf{Im}(f)$. Given a set $\mathcal{A}$ of annotated trees, $\mathsf{crop}(\mathcal{A}) := \bigcup_{\hat{T} \in \mathcal{A}} \mathsf{crop}(\hat{T})$.

Representation operation. Given an annotated tree $\hat{T} = (T, r, h, (R, B, \phi), f)$, the *representation operation*, denoted by $\mathsf{rep}(\hat{T})$, outputs the annotated tree $\hat{T}' = (T, r, h, (R', B, \phi), f)$ constructed as follows. For each $v \in \mathsf{Im}(f)$, let $B_v := \phi^{-1} \circ f^{-1}(v)$, let $\sigma_v : f^{-1}(v) \to [|B_v|]$ be a bijective function, and let R_v be the union of the connected components of R containing B_v. If there is a node $v \in \mathsf{Im}(f)$ such that $R_v \notin \mathsf{exc}(\mathcal{F})$, then $R' := K_{s_{\mathcal{F}}}$. Otherwise, for each $v \in \mathsf{Im}(f)$, let $(R'_v, B_v, \sigma_v \circ \phi|_{B_v}) \in \mathcal{R}^{|B_v|}_{\ell_{\mathcal{F}}}$ be the representative of $(R_v, B_v, \sigma_v \circ \phi|_{B_v})$ for the equivalence relation $\equiv_{\ell_{\mathcal{F}}}$. Then $R' = \bigcup_{v \in \mathsf{Im}(f)} R'_v$. Intuitively, if there is a node $v \in \mathsf{Im}(f)$ such that $R_v \notin \mathsf{exc}(\mathcal{F})$, to store this information it suffices to set $R' := K_{s_{\mathcal{F}}}$, while otherwise, we keep for each R_v (in fact, for the boundaried version of R_v) its representative.

An example of the crop and representation operation is give in Figure 8.2. Observe that $\mathsf{rep}(\hat{\mathbb{1}}) = \hat{\mathbb{1}}$ since this is a minimum-sized representative and since we make the assumption that $\mathcal{F}$ is non-trivial. Given a set $\mathcal{A}$ of annotated trees, $\mathsf{rep}(\mathcal{A}) := \bigcup_{\hat{T} \in \mathcal{A}} \mathsf{rep}(\hat{T})$.

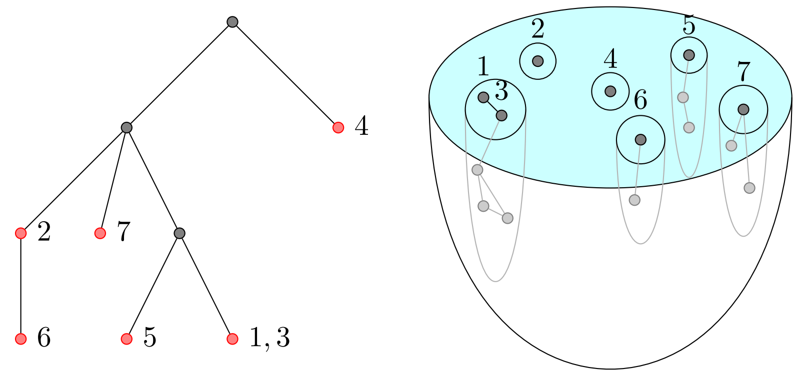

Figure 8.2: The crop and representation operation applied to the annotated tree of Figure 8.1. The unlabeled leaves of the tree are iteratively removed. A representative of each component attached to the boundary of the boundaried graph is kept.

Filter operation. Given a set $\mathcal{A}$ of annotated trees and a positive integer k, the *filter operation*, denoted by filter_k, outputs the set of annotated trees in $\mathcal{A}$ with height at most k.

Note that the crop, representation, and filter operations are commutative since they do not modify the same objects. For more simplicity, we define $\mathcal{M}_k = \mathsf{filter}_k \circ \mathsf{rep} \circ \mathsf{crop}$. We stress that $\mathcal{M}_k$ is an operation acting on *sets* of annotated trees.

8.5.2 Characteristic of a boundaried graph

In this subsection we define the *characteristic* of a boundaried graph that shall be computed by the dynamic program in Subsection 8.5.4. This characteristic will consist of a set of annotated trees with some additional properties. In order to present this definition, we first define the complete characteristic of a boundaried graph, that is a slightly more complicated way to see $\mathcal{F}$-elimination trees with some distinguished nodes.

Complete characteristic of a boundaried graph. Given a connected boundaried graph $\mathbf{G} = (G, X, \rho)$, the *complete characteristic* of $\mathbf{G}$, denoted by $\mathsf{char}^*(\mathbf{G})$, is the set of annotated trees $\hat{T} = (T, r, h, \mathbf{R}, f)$ such that

- $|\mathsf{Im}(f)| = |X|$,
- there exists a function $\chi : V(T) \to 2^{V(G)}$ such that (T, χ, r) is an $\mathcal{F}$-elimination tree of G and for $x \in X$, $x \in \chi \circ f \circ \rho(x)$,
- there exists an isomorphism σ between $\mathbf{R}$ and $(\bigcup_{v \in \mathsf{Im}(f)} G[\chi(v)], X, \rho)$, and
- h is the height function $\mathsf{height}_{T,r}$.

(χ, σ) is called the *witness pair* of $\hat{T}$ with respect to $\mathbf{G}$. Since (T, χ, r) is an $\mathcal{F}$-elimination tree, it is straightforward to see that for any boundaried graph $\mathbf{G}$ with underlying graph G, the minimum height of an annotated tree in $\mathsf{char}^*(\mathbf{G})$ is $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G)$.

Characteristic of a boundaried graph. Let $\mathbf{G} = (G, X, \rho)$ be a boundaried graph and k be an integer. The *characteristic* of $\mathbf{G}$, denoted by $\mathsf{char}_k(\mathbf{G})$, is the set $\mathcal{M}_k(\mathsf{char}^*(\mathbf{G}))$.

Lemma 8.5.5. *Given a boundaried graph $\mathbf{G} = (G, X, \rho)$ with $X \neq \emptyset$ and an integer k, the elimination distance of G to $\mathsf{exc}(\mathcal{F})$ is the minimum height of an annotated tree in $\mathsf{char}_k(\mathbf{G})$ if $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \leq k$, and $\mathsf{char}_k(\mathbf{G}) = \emptyset$ otherwise.*

Proof. Let $\mathbf{G} = (G, X, \rho)$ be a boundaried graph with $X \neq \emptyset$ and let $\hat{T} \in \mathsf{char}^*(\mathbf{G})$. Since $X \neq \emptyset$, the crop operation will not remove the root of the underlying tree of $\hat{T}$. Moreover, neither the crop operation nor the representation operation change the height of the nodes that stay in the tree. So the height of $\mathsf{rep} \circ \mathsf{crop}(\hat{T})$ is equal to the height of $\hat{T}$. □

We now prove that the size of the characteristic of a boundaried graph is upper-bounded by a function of its boundary size and k.

Lemma 8.5.6. *There exists a function $f_{8.5.6} : \mathbb{N}^2 \to \mathbb{N}$ such that, given two integers k and w, if $\mathbf{G} = (G, X, \rho)$ is a boundaried graph with $|X| \leq w$, then $|\mathsf{char}_k(\mathbf{G})| \leq f_{8.5.6}(w, k)$. Moreover, $f_{8.5.6}(w, k) = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(w \cdot k + w \log w)}$.*

Proof. Let $\hat{T} = (T, r, h, \mathbf{R}, f)$ be an annotated tree in $\mathsf{char}_k(\mathbf{G})$. Let $l := |X|$. Let $x_1, ..., x_l$ be an ordering of the nodes in $\mathsf{Im}(f)$ such that if $x_i \in \mathsf{Anc}_{T,r}(x_j)$, then $i < j$. Without loss of generality, we suppose that $f(i) = x_i$ for $i \in [l]$ (this is true up to a permutation of $[l]$). Let f_i be the restriction of f to $[i]$, let T_i be the tree obtained from T by iteratively removing the leaves not in $\mathsf{Im}(f_i)$, and let h_i be the restriction of h to $V(T_i)$. Note that $\mathsf{Im}(h_i) \subseteq [0, k]$ and T_i is a tree with at most i leaves because the leaves of T are in $\mathsf{Im}(f_i)$ and $|\mathsf{Im}(f_i)| = i$. So T_i has at most $i \cdot (k + 1)$ nodes.

We set (T_0, h_0, f_0) to be the empty triple. Let us bound the number of triples (T_i, h_i, f_i) that can be constructed from $(T_{i-1}, h_{i-1}, f_{i-1})$ for $i \in [l]$. For $i \in [l]$, we can construct (T_i, h_i, f_i) from $(T_{i-1}, h_{i-1}, f_{i-1})$ by choosing a node of T_{i-1} (if it exists) and adding a path of length at most k with leaf x_i (all nodes in this path are new, except from x_i). We consider the function h_i that has the same values as h_{i-1} on $V(T_{i-1})$ and values in $[0, k]$ on the new path such that the value of h_i strictly increases from a leaf to the root r. Observe that the value of h_i on the new path is a subset of $2^{[k+1]}$. Therefore, the number of different triples (T, h, f) is at most

$$\prod_{i=1}^{w} i(k+1)2^{k+1} = w!(k+1)^w 2^{w(k+1)} \leq 2^{w \log w + w \log(k+1) + w(k+1)}.$$

Since $\hat{T} = (T, r, h, \mathbf{R}, f)$ is an annotated tree in $\mathsf{char}_k(\mathbf{G})$, it holds that $\mathbf{R}$ is the union of l representatives $\mathbf{R}_i \in \mathcal{R}^{w_i}_{\ell_{\mathcal{F}}}$ for $i \in [l]$ where $l = |\mathsf{Im}(f)|$, such that $\sum_{i=1}^{l} w_i = l \leq w$. By Proposition 4.4.2, $|\mathcal{R}^{w_i}_{\ell_{\mathcal{F}}}| = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(w_i \log w_i)}$. So the number of ways to construct $\mathbf{R}$ is bounded by

$$\prod_{i=1}^{l} 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(w_i \log w_i)} = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(\sum_{i=1}^{l} w_i \log w)} = 2^{\mathcal{O}_{\ell_{\mathcal{F}}}(w \log w)}.$$

Hence, we obtain the desired result. □

8.5.3 The procedures

We define here the procedures that will be used in the dynamic programming algorithm. Given a nice tree decomposition $\mathcal{T}$ of a graph G, we want to define forget, introduce, and join procedures to obtain the characteristic of G_v for each internal node v in $\mathcal{T}$, given the characteristics of $G_{v'}$ for each child v' of v. Before defining the procedures for the characteristics, we define the procedures for the complete characteristics.

Forget procedure

With the forget procedure, given the characteristic of a boundaried graph, we want to compute the characteristic of the boundaried graph obtained by removing a vertex from the boundary.

Complete forget procedure. The complete forget procedure applied on the annotated tree $(T, r, h, \mathbf{R}, f)$ corresponds to removing the vertex with the largest label from the boundary of $\mathbf{R}$. More formally, given an annotated tree $\hat{T} = (T, r, h, (R, B, \phi), f)$, the *complete forget procedure*, denoted by $\mathsf{forget}^*(\hat{T})$, outputs the annotated tree $\hat{T}' = (T, r, h, (R, B', \phi|_{B'}), f|_{[|B'|]}))$, where $B' = B \setminus \phi^{-1}(|B|)$. Given a set $\mathcal{A}$ of annotated trees, $\mathsf{forget}^*(\mathcal{A}) := \bigcup_{\hat{T} \in \mathcal{A}} \mathsf{forget}^*(\hat{T})$.

Lemma 8.5.7. *Let G be a graph, (T, β, r) be a nice tree decomposition of G, v be a forget node of T with child v' and forgotten vertex x, $\rho : \beta(v) \to [|\beta(v)|]$ be a bijection, and $\rho' := \rho \cup (x \mapsto |\beta(v')|)$. Let $\mathcal{A} := \mathsf{char}^*(G_v, \beta(v), \rho)$ and $\mathcal{A}' := \mathsf{char}^*(G_{v'}, \beta(v'), \rho')$. Then $\mathcal{A} = \mathsf{forget}^*(\mathcal{A}')$.*

Proof. Let $\hat{T} \in \mathsf{forget}^*(\mathcal{A}')$. There exists $\hat{T}' \in \mathcal{A}'$ with witness pair (χ', σ') such that $\mathsf{forget}^*(\hat{T}') = \hat{T}$. Let (T, r) (resp. (T', r)) be the rooted tree of $\hat{T}$ (resp. $\hat{T}'$). Note that $\mathsf{height}_{T,r} = \mathsf{height}_{T',r}$. Thus, it is easy to see that (χ', σ') also witnesses that $\hat{T} \in \mathcal{A}$. Conversely, let $\hat{T} = (T, r, h, (R, X, \phi), f) \in \mathcal{A}$ with witness pair (χ, σ). Let $u := \chi^{-1}(x)$, $w := \sigma^{-1}(x)$, and $t = |X| + 1$. Let $\hat{T}' := (T, r, h, (R, X \cup \{w\}, \phi \cup (w \mapsto t)), f \cup (t \mapsto u))$. Then $\mathsf{forget}^*(\hat{T}') = \hat{T}$ and the pair (χ, σ) also witnesses that $\hat{T}' \in \mathcal{A}'$, so $\hat{T} \in \mathsf{forget}^*(\mathcal{A}')$. □

Forget procedure. Given an annotated tree $\hat{T}$, the *forget procedure*, denoted by $\mathsf{forget}(\hat{T})$, outputs $\mathsf{rep} \circ \mathsf{crop} \circ \mathsf{forget}^*(\hat{T})$. See Figure 8.3 for an illustration. Note that we do not apply the filter operation since the height does not change under the forget procedure. Given a set $\mathcal{A}$ of annotated trees, $\mathsf{forget}(\mathcal{A})$ outputs $\bigcup_{\hat{T} \in \mathcal{A}} \mathsf{forget}(\hat{T})$.

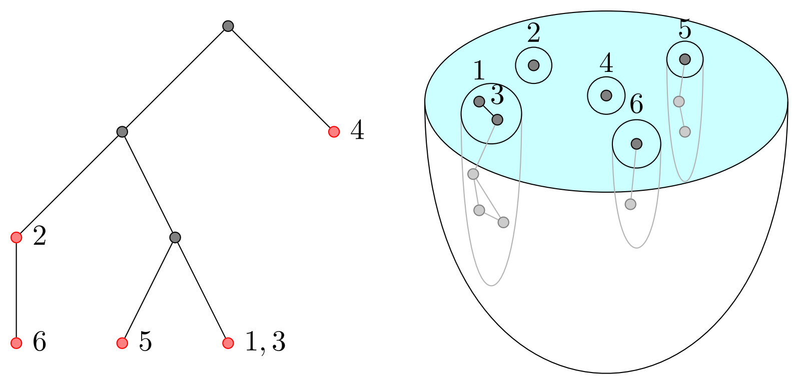

Figure 8.3: The forget procedure applied to the annotated tree of Figure 8.1.

Lemma 8.5.8. *Let G be a graph, k be an integer, (T, β, r) be a nice tree decomposition of G, v be a forget node of T with child v' and forgotten vertex x, $\rho : \beta(v) \to [|\beta(v)|]$ be a bijection, and $\rho' := \rho \cup (x \mapsto |\beta(v')|)$. Let $\mathcal{A} := \mathsf{char}_k(G_v, \beta(v), \rho)$ and $\mathcal{A}' := \mathsf{char}_k(G_{v'}, \beta(v'), \rho')$. Then $\mathcal{A} = \mathsf{forget}(\mathcal{A}')$.*

Proof. Let $\mathcal{D} := \mathsf{char}^*(G_v, \beta(v), \rho)$ and $\mathcal{D}' := \mathsf{char}^*(G_{v'}, \beta(v'), \rho')$. According to Lemma 8.5.7, $\mathcal{D} = \mathsf{forget}^*(\mathcal{D}')$. The representation and the crop operation do not change the labels of an annotated tree, while the complete forget procedure only changes the labels of the input annotated tree and it does not change its height. Thus, $\mathsf{rep} \circ \mathsf{forget}^* = \mathsf{rep} \circ \mathsf{forget}^* \circ \mathsf{rep}$, $\mathsf{crop} \circ \mathsf{forget}^* = \mathsf{crop} \circ \mathsf{forget}^* \circ \mathsf{crop}$, and $\mathsf{filter}_k \circ \mathsf{forget}^* = \mathsf{forget}^* \circ \mathsf{filter}_k$. Since the three operations are commutative, it follows that $\mathcal{M}_k \circ \mathsf{forget}^* = \mathsf{rep} \circ \mathsf{crop} \circ \mathsf{forget}^* \circ \mathcal{M}_k = \mathsf{forget} \circ \mathcal{M}_k$. Hence, $\mathcal{A} = \mathcal{M}_k(\mathcal{D}) = \mathcal{M}_k \circ \mathsf{forget}^*(\mathcal{D}') = \mathsf{forget}(\mathcal{M}_k(\mathcal{D}')) = \mathsf{forget}(\mathcal{A}')$. ☐

Introduce procedure

With the introduce procedure, given the characteristic of a boundaried graph and a set I of labels from the boundary, we want to compute the characteristic of the boundaried graph obtained by adding a new vertex to the boundary, which is adjacent to the nodes with a label in I.

Diamond-introduce operation. Let (T, r) be a rooted tree, w be an integer, $f : [w] \to V(T)$ be a function, and I be a subset of $[w]$. $(T, r, f)\Diamond_{\mathsf{intr}} I$ is defined as the set of all pairs (T', r', f') such that:

1. (T', r') is a rooted tree,
2. $V(T') = V(T) \cup \{u\}$ for some new node u,
3. $f' = f \cup (w + 1 \mapsto u)$,
4. if $v_1 \in V(T)$ and $v_2 \in \mathsf{Anc}_{T,r}(v_1)$, then $v_2 \in \mathsf{Anc}_{T',r'}(v_1)$,
5. if $v \in f(I)$, then $v \in \mathsf{Anc}_{T,r}(u) \cup \mathsf{Desc}_{T,r}(u)$, and
6. $T'_u \cap f(I) \neq \emptyset$, or $u \in \mathsf{Leaf}(T', r')$ and $\mathsf{Par}_{T',r'}(u) \in f(I)$.

This operation corresponds to introducing a new node u in T so that u has ancestor-descendant relations with the nodes labeled by a label in I. The last item, which states that u either has a descendant in $f(I)$ or is a leaf and its parent belongs to $f(I)$, is a property needed to ensure connectivity and allows the application of the crop operation in Lemma 8.5.9 and Lemma 8.5.10.

Let (T, r) be a rooted tree, K be a subset of $V(T)$, and $h : K \to \mathbb{N}$. We define the function $\mathsf{update}_{T,r,K}(h) : V(T) \to \mathbb{N}$, that maps every $v \in V(T)$ to the integer

$$\mathsf{update}_{T,r,K}(h)(v) = \max\{h(v), 1 + \max_{c \in \mathsf{Ch}(v)} \{\mathsf{update}_{T,r,K}(h)(c)\}\},$$

where we suppose that $h(v) = 0$ if $v \notin K$ and $\mathsf{update}_{T,r,K}(h)(c) = -1$ if v is childless. Let (T', r') be a rooted tree with $V(T') = K$ and such that the ancestor-descendant relationship between the nodes of K is the same in (T, r) and (T', r'). Then we can observe that $\mathsf{update}_{T,r,K}(\mathsf{height}_{T',r'}) = \mathsf{height}_{T,r}$.

Complete introduce procedure. Let $\hat{T} = (T, r, h, (R, X, \phi), f)$ be an annotated tree and let I be a set of labels in $[|X|]$. The complete introduce procedure corresponds to adding a new vertex v to the boundary X of the boundaried graph (R, X, ϕ) of an annotated tree $(T, r, h, (R, X, \phi), f)$, such that v is adjacent to the nodes with a label in I and it is either mapped, via $f \circ \phi$, to an already existing leaf of T (item (a) below) or to a new node of T (item (b) below).

More formally, given an annotated tree $\hat{T} = (T, r, h, (R, X, \phi), f)$, a set $I \subseteq [|X|]$ of labels, the *complete introduce procedure*, denoted by $\mathsf{intr}^*(\hat{T}, I)$, outputs a set $\mathcal{A}$ of annotated trees constructed as follows. For each $(T', r', f') \in (T, r, f)\Diamond_{\mathsf{intr}} I$, let $w = |\mathsf{Im}(f')|$, $u := f'(w)$, and $u' := \mathsf{Par}_{T',r'}(u)$ (or $u' := u$ if u is the root). We add a new vertex v to R and we set $X' := X \cup \{v\}$ and $\phi' := \phi \cup (v \mapsto w)$. Let $(R_{u'}, X_{u'}, \phi_{u'})$ be the part of the representative R corresponding to u', as defined for the representation operation (i.e., $R_{u'}$ is the union of the connected components of R containing $\phi^{-1} \circ f^{-1}(u')$, $X_{u'} = \phi^{-1} \circ f^{-1}(u')$, $\phi_{u'} = \sigma_{u'} \circ \phi|_{X_{u'}}$ and $\sigma_{u'} : f^{-1}(u') \to [|X_{u'}|]$ is a bijective function).

(a) *If $h(u') = 0$:* We set $f'' := f \cup (w \mapsto u')$ and $R' := (V(R) \cup \{v\}, E(R) \cup E(\{v\}, \phi^{-1}(I) \cap X_{u'}))$ and we add $(T, r, h, (R', X', \phi'), f'')$ to $\mathcal{A}$ if the connected component of R' containing v belongs to $\mathsf{exc}(\mathcal{F})$.

(b) *If $u' \notin \mathsf{Im}(f)$, or if $u' \in \mathsf{Im}(f)$ and $|V(R_{u'})| = 1$:* We set $R' := (V(R) \cup \{v\}, E(R))$ and $h' := \mathsf{update}_{T',r',V(T)}(h)$ and we add $(T', r', h', (R', X,' \phi'), f')$ to $\mathcal{A}$.

Note that this is not a dichotomy: if both criteria are fulfilled, then we apply both cases. Given a set $\mathcal{A}$ of annotated trees, we define $\mathsf{intr}^*(\mathcal{A}, I) := \bigcup_{\hat{T} \in \mathcal{A}} \mathsf{intr}^*(\hat{T}, I)$.

Lemma 8.5.9. *Let G be a graph, (T, β, r) be a nice tree decomposition of G, v be an introduce node of T with child v' and introduced vertex x, $\rho' : \beta(v) \to [|\beta(v)|]$ be a bijection, $\rho := \rho' \cup (x \mapsto |\beta(v)|)$, and $I := \rho(N_{G_v}(x))$. Let $\mathcal{A} := \mathsf{char}^*(G_v, \beta(v), \rho)$ and $\mathcal{A}' := \mathsf{char}^*(G_{v'}, \beta(v'), \rho')$. Then $\mathcal{A} = \mathsf{intr}^*(\mathcal{A}', I)$.*

Proof. Let $\hat{T} = (T, r, h, (R, X, \phi), f) \in \mathsf{intr}^*(\mathcal{A}', I)$. There is $\hat{T}' = (T', r', h', \mathbf{R}', f') \in \mathcal{A}'$ with witness pair (χ', σ') such that $\hat{T} \in \mathsf{intr}^*(\hat{T}', I)$. Let $\chi := \chi' \cup (u \mapsto x)$ and $\sigma = \sigma' \cup (w \to x)$ where $u := f(|X|)$ and $w := \phi^{-1}(|X|)$.

Suppose that we are in case (a). Thus, $h = h' = \mathsf{height}_{T',r'} = \mathsf{height}_{T,r}$. By the construction of $\mathbf{R}$ from $\mathbf{R}'$, since x is only adjacent in G_v to the nodes with a label in I, σ is an isomorphism between $\mathbf{R}$ and $(\bigcup_{w \in \mathsf{Im}(f)} G_v[\chi(w)], \beta(v), \rho)$. Moreover, $T = T'$ so (T, χ, r) is an $\mathcal{F}$-elimination tree of G_v. Hence, (χ, σ) witnesses that $\hat{T} \in \mathcal{A}$.

Suppose now that we are in case (b). Notice that in this case

$$h = \mathsf{update}_{T,r,V(T')}(h') = \mathsf{update}_{T,r,V(T')}(\mathsf{height}_{T',r'}) = \mathsf{height}_{T,r}.$$

It is easy to see that σ is an isomorphism between $\mathbf{R}$ and $(\bigcup_{w \in \mathsf{Im}(f)} G_v[\chi(w)], \beta(v), \rho)$. Moreover, as $\hat{T} \in \mathsf{intr}^*(\hat{T}', I)$, it holds that $(T, r, f) \in (T', r', f') \diamondsuit_{\mathsf{intr}} I$ and therefore (T, r) keeps the ancestor-descendant relations from (T', r') (item 4 of the $\diamondsuit_{\mathsf{intr}}$ operation), while adding ancestor-descendant relations between the new node of T and the nodes labeled by I (item 5), and guaranteeing the connectivity of $G_v[\chi(T_w)]$ for each node $w \in V(T)$ (item 6). Hence, (χ, σ) witnesses that $\hat{T} \in \mathcal{A}$.

Conversely, let $\hat{T} = (T, r, h, \mathbf{R}, f) \in \mathcal{A}$ with witness pair (χ, σ). If $\chi(\chi^{-1}(x)) = \{x\}$, then let T' be the tree obtained from T by removing $u := \chi^{-1}(x)$ and adding edges between the parent of u and the children of u, if any. Otherwise, set $T' := T$. Let h' be the height function of T' and r' be the root of T'. Let (R', X', ϕ') be the boundaried graph obtained from $\mathbf{R}$ by removing $\sigma^{-1}(x)$. Let $f' := f|_{X'}$. Then the functions obtained from χ and σ after restricting their image to $V(G) \setminus x$ witness that $\hat{T}' = (T', r', h', (R', X', \phi'), f') \in \mathcal{A}'$. To show that $\hat{T} \in \mathsf{intr}^*(\hat{T}', I)$, the only non-trivial part is to prove that either $T_u \cap f(I) \neq \emptyset$, or $u \in \mathsf{Leaf}(T, r)$ and $\mathsf{Par}_{T,r}(u) \in f(I)$ (item 6 of the $\diamondsuit_{\mathsf{intr}}$ operation). Suppose that $T_u \cap f(I) = \emptyset$. We know that x is exactly adjacent to $\rho^{-1}(I)$ in G_v since it is an introduce vertex. Moreover, (T, χ, r) is an $\mathcal{F}$-elimination tree of G_v, so $G_v[\chi(T_w)]$ is connected for all $w \in V(T)$. In particular, $G_v[\chi(T_u)]$ is connected, but $x \in G_v[\chi(T_u)]$ and $\rho^{-1}(I) \notin G_v[\chi(T_u)]$ because $T_u \cap f(I) = \emptyset$. Thus, we must have $G_v[\chi(T_u)] = \{x\}$, and so $u \in \mathsf{Leaf}(T, r)$. Since $G_v[\chi(T_{\mathsf{Par}_{T,r}(u)})]$ is also connected, it implies that $\chi(\mathsf{Par}_{T,r}(u))$ and x are connected, so $\mathsf{Par}_{T,r}(u) \in f(I)$. Hence, $\hat{T} \in \mathsf{intr}^*(\mathcal{A}', I)$. □

Introduce procedure. Given an annotated tree $\hat{T}$, a set I of labels, and a positive integer k, the *introduce procedure*, denoted by $\mathsf{intr}_k(\hat{T}, I)$, outputs $\mathsf{filter}_k \circ \mathsf{rep} \circ \mathsf{intr}^*(\hat{T}, I)$. Examples of the introduce procedure can be found in Figure 8.4. Note that we do not apply the crop operation, since the complete introduction procedure applied to a cropped annotated tree outputs a cropped annotated tree. Given a set $\mathcal{A}$ of annotated trees, $\mathsf{intr}_k(\mathcal{A}, I)$ outputs $\bigcup_{\hat{T}\in\mathcal{A}} \mathsf{intr}_k(\hat{T}, I)$.

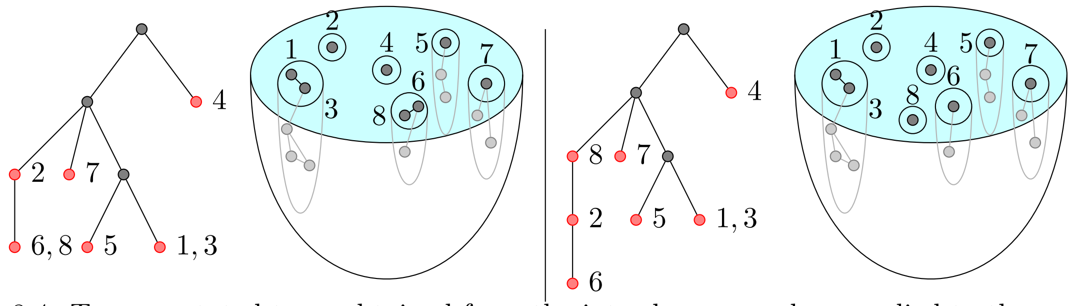

Figure 8.4: Two annotated trees obtained from the introduce procedure applied to the annotated tree of Figure 8.1 with $I = \{2, 6\}$.

Lemma 8.5.10. *Let G be a graph, (T, β, r) be a nice tree decomposition of G, v be an introduce node of T with child v' and introduced vertex x, $\rho' : \beta(v) \to [|\beta(v)|]$ be a bijection, $\rho := \rho' \cup (x \mapsto |\beta(v)|)$, and $I := \rho(N_{G_v}(x))$. Let k be an integer, $\mathcal{A} := \mathsf{char}_k(G_v, \beta(v), \rho)$, and $\mathcal{A}' := \mathsf{char}_k(G_{v'}, \beta(v'), \rho')$. Then $\mathcal{A} = \mathsf{intr}_k(\mathcal{A}', I)$.*

Proof. Let $\mathcal{D} := \mathsf{char}^*(G_v, \beta(v), \rho)$ and $\mathcal{D}' := \mathsf{char}^*(G_{v'}, \beta(v'), \rho')$. According to Lemma 8.5.9, $\mathcal{D} = \mathsf{intr}^*(\mathcal{D}', I)$.

The crop operation acts on the tree T of an annotated tree $(T, r, h, \mathbf{R}, f)$ to remove the leaves that are not in $\mathsf{Im}(f)$. By item 6 of the $\diamondsuit_{\mathsf{intr}}$ operation, the complete introduce procedure may only add a new node in T above a node in $\mathsf{Im}(f)$, or add a leaf to T with a parent node in $\mathsf{Im}(f)$. Hence, the new node may only be added to the cropped tree. Since the new vertex is also in the boundary, it implies that, given a set $\mathcal{C}$ of annotated trees, $\mathsf{crop} \circ \mathsf{intr}^*(\mathcal{C}, I) = \mathsf{intr}^*(\mathsf{crop}(\mathcal{C}), I)$.

The representation operation acts on $\mathbf{R}$ to replace it by a boundaried graph whose connected components are representatives of the connected components of $\mathbf{R}$. The complete introduce procedure adds a new vertex in the boundary of $\mathbf{R}$ that is adjacent to boundary vertices only. Hence, $\mathsf{rep} \circ \mathsf{intr}^*(\mathcal{C}, I) = \mathsf{rep} \circ \mathsf{intr}^*(\mathsf{rep}(\mathcal{C}), I)$.

Moreover, the complete introduce procedure can only increase the height of an annotated tree. Therefore, $\mathsf{filter}_k \circ \mathsf{intr}^*(\mathcal{C}, I) = \mathsf{filter}_k \circ \mathsf{intr}^*(\mathsf{filter}_k(\mathcal{C}), I)$.

Since the three operations are commutative, we have that $\mathcal{M}_k \circ \mathsf{intr}^*(\mathcal{C}, I) = \mathsf{filter}_k \circ \mathsf{rep} \circ \mathsf{intr}^*(\mathcal{M}_k(\mathcal{C}), I) = \mathsf{intr}_k(\mathcal{M}_k(\mathcal{C}), I)$. Therefore, $\mathcal{A} = \mathcal{M}_k(\mathcal{D}) = \mathcal{M}_k \circ \mathsf{intr}^*(\mathcal{D}', I) = \mathsf{intr}_k(\mathcal{M}_k(\mathcal{D}'), I) = \mathsf{intr}_k(\mathcal{A}', I)$. $\square$

For the introduce procedure, we finally prove that it can generate a bounded number of annotated trees.

Lemma 8.5.11. *Let w and k be two positive integers, let $\mathbf{G}$ be a w-boundaried graph, let $I \subseteq [w]$, and let $\hat{T} \in \mathsf{char}_k(\mathbf{G})$. Then $|\mathsf{intr}_k(\hat{T}, I)| = \mathcal{O}(w \cdot k)$.*

Proof. Let $\hat{T} = (T, r, h, \mathbf{R}, f)$. Let us show that $|(T, r, f)\diamondsuit_{\mathsf{intr}} I| = \mathcal{O}(w \cdot k)$. Since $\mathsf{Leaf}(T, r) \subseteq \mathsf{Im}(f)$, T has at most w leaves. Thus, T has at most $w \cdot (k + 1)$ nodes. The $\diamondsuit_{\mathsf{intr}}$ operation consists in adding a new node to T, either as the new parent of a node, or as a new leaf of T. Therefore, $|(T, r, f)\diamondsuit_{\mathsf{intr}} I| \leq 2w \cdot (k + 1)$.

Let $(T', r', f') \in (T, r, f)\diamondsuit_{\mathsf{intr}} I$. In the introduction procedure, we obtain at most two annotated trees from (T', r', f'). Therefore, $|\mathsf{intr}_k(\hat{T}, I)| \leq 4w \cdot (k+1)$. $\square$

Join procedure

With the join procedure, given the characteristic of two boundaried graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ that are compatible, we want to compute the characteristic of $\mathbf{G}_1 \bigoplus \mathbf{G}_2$.

Diamond-join operation. Let (T_1, r_1) and (T_2, r_2) be two rooted trees, w be an integer, and $f_1 : [w] \to V(T_1)$ and $f_2 : [w] \to V(T_2)$ be two functions such that $\{f_1^{-1}(u) \mid u \in \mathsf{Im}(f_1)\} = \{f_2^{-1}(u) \mid u \in \mathsf{Im}(f_2)\}$. Thanks to this equality, we can identify f_1 with f_2 and say that $V(T_1) \cap V(T_2) = \mathsf{Im}(f_1)$. We define $(T_1, r_1, f_1)\diamondsuit_{\mathsf{join}}(T_2, r_2, f_2)$ as the set of all pairs (T, r, f) such that:

1. $f = f_1 = f_2$,
2. (T, r) is a rooted tree,
3. $V(T) = V(T_1) \cup V(T_2)$,
4. for $i \in \{1, 2\}$, if $u \in V(T_i)$ and $v \in \mathsf{Anc}_{T_i, r_i}(u)$, then $v \in \mathsf{Anc}_{T,r}(u)$, and
5. for every $v \in \mathsf{Leaf}(T, r)$ and every $w \in \mathsf{Anc}_{T,r}(v)$, if $V(vTw) \cap \mathsf{Im}(f) = \emptyset$, then there is an $i \in \{1, 2\}$ such that $vTw = vT_iw$.

The last item, which states that a branch of T that does not intersect $\mathsf{Im}(f)$ either belongs to T_1 or T_2, is a property needed to ensure connectivity and allows the application of the crop operation in Lemma 8.5.12 and Lemma 8.5.13.

Let (T, r) be a rooted tree, and for $i \in \{1, 2\}$, let $K_i \subseteq V(T)$ and $h_i : K_i \to \mathbb{N}$. $\mathsf{update}_{T,r,K_1,K_2}(h_1, h_2)$ is the function that maps $v \in V(T)$ to the maximum of $h_1(v)$, $h_2(v)$, and $\max_{c \in \mathsf{Ch}(v)}\{1+\mathsf{update}_{T,r,K_1,K_2}(h_1, h_2)(c)\}$, when they are defined. For $i \in \{1, 2\}$, let (T_i, r_i) be a rooted tree with $V(T_i) = K_i$ and such that the ancestor-descendant relationship between the nodes of K_i is the same in T and T_i. Then we can observe that $\mathsf{update}_{T,r,K_1,K_2}(\mathsf{height}_{T_1,r_1}, \mathsf{height}_{T_2,r_2}) = \mathsf{height}_{T,r}$.

Complete join procedure. The join procedure corresponds to "merging" two annotated trees whose intersection is exactly their boundary. More formally, given two annotated trees $\hat{T}_1 = (T_1, r_1, h_1, \mathbf{R}_1, f_1)$ and $\hat{T}_2 = (T_2, r_2, h_2, \mathbf{R}_2, f_2)$, the *complete join procedure*, denoted by $\mathsf{join}(\hat{T}_1, \hat{T}_2)$, outputs a set $\mathcal{A}$ of annotated trees constructed as follows. Initially, $\mathcal{A}$ is empty. If $\mathbf{R}_1$ and $\mathbf{R}_2$ are compatible (i.e., such that $\{f_1^{-1}(u) \mid u \in \mathsf{Im}(f_1)\} = \{f_2^{-1}(u) \mid u \in \mathsf{Im}(f_2)\}$) and such that $\mathsf{Ker}(h_1 \circ f_1) = \mathsf{Ker}(h_2 \circ f_2)$, then for $(T, r, f) \in (T_1, r_1, f_1)\diamondsuit_{\mathsf{join}}(T_2, r_2, f_2)$, let $h := \mathsf{update}_{T,r,V(T_1),V(T_2)}(h_1, h_2)$. Let $(R, X, \rho) := \mathbf{R}_1 \bigoplus \mathbf{R}_2$. If each connected component of R belongs to $\mathsf{exc}(\mathcal{F})$, then we add $(T, r, h, \mathbf{R}_1 \bigoplus \mathbf{R}_2, f)$ to $\mathcal{A}$. Given two sets $\mathcal{A}_1$ and $\mathcal{A}_2$ of annotated trees, we set $\mathsf{join}^*(\mathcal{A}_1, \mathcal{A}_2) := \bigcup_{\hat{T}_1 \in \mathcal{A}_1, \hat{T}_2 \in \mathcal{A}_2} \mathsf{join}^*(\hat{T}_1, \hat{T}_2)$.

Lemma 8.5.12. *Let G be a graph, (T, β, r) be a nice tree decomposition of G, v be a join node of T with children v_1 and v_2, and $\rho : \beta(v) \to [|\beta(v)|]$ be a bijection. Let $\mathcal{A} := \mathsf{char}^*(G_v, \beta(v), \rho)$, $\mathcal{A}_1 := \mathsf{char}^*(G_{v_1}, \beta(v_1), \rho)$, and $\mathcal{A}_2 := \mathsf{char}^*(G_{v_2}, \beta(v_2), \rho)$. Then $\mathcal{A} = \mathsf{join}^*(\mathcal{A}_1, \mathcal{A}_2)$.*

Proof. Let $\hat{T} = (T, r, h, (R, X, \phi), f) \in \mathsf{join}^*(\mathcal{A}_1, \mathcal{A}_2)$. There is $\hat{T}_1 = (T_1, r_1, h_1, \mathbf{R}_1, f_1) \in \mathcal{A}_1$ and $\hat{T}_2 = (T_2, r_2, h_2, \mathbf{R}_2, f_2) \in \mathcal{A}_2$ such that $\hat{T} \in \mathsf{join}^*(\hat{T}_1, \hat{T}_2)$ with witness pair (χ_1, σ_1) and (χ_2, σ_2), respectively, such that $\chi_{1|\mathsf{Im}(f)} = \chi_{2|\mathsf{Im}(f)}$ and $\sigma_{1|X} = \sigma_{2|X}$. Note that

$$h := \mathsf{update}_{T,r,V(T_1),V(T_2)}(h_1, h_2) = \mathsf{update}_{T,r,V(T_1),V(T_2)}(\mathsf{height}_{T_1,r_1}, \mathsf{height}_{T_2,r_2}) = \mathsf{height}_{T,r}.$$

Let $(\chi, \sigma) = (\chi_1 \cup \chi_2, \sigma_1 \cup \sigma_2)$.

It is easy to see that σ is an isomorphism between $\mathbf{R}$ and $(\bigcup_{w \in \mathsf{Im}(f)} G[\chi(w)], \beta(v), \rho)$. If $uv \in E(G_v)$, since $G_{v_1} - \beta(v)$ and $G_{v_2} - \beta(v)$ are not connected, then there is $i \in \{1,2\}$ such that $\chi_i(u) \in \mathsf{Anc}_{T,r}(\chi_i(v)) \cup \mathsf{Desc}_{T,r}(\chi_i(v))$ holds, so $\chi(u) \in \mathsf{Anc}_{T,r}(\chi(v)) \cup \mathsf{Desc}_{T,r}(\chi(v))$ due to item 4 of the $\diamondsuit_{\mathsf{join}}$ operation.

Moreover, item 5 of the $\diamondsuit_{\mathsf{join}}$ operation ensures the connectivity of $G_v[\chi(T_w)]$ for each $w \in V(T)$. Indeed, let $i \in \{1,2\}$ be such that $w \in V(T_i)$. Suppose towards a contradiction that $G_v[\chi(T_w)]$ is not connected. Note that it implies that $w \notin \mathsf{Im}(f)$, because $\mathsf{Im}(f) \subseteq V(T_1) \cup V(T_2)$ and (T_1, χ_1, r_1) and (T_2, χ_2, r_2) are $\mathcal{F}$-elimination trees of G_{v_1} and G_{v_2}, respectively, so $G_{v_1}[\chi_1((T_1)_w)]$ and $G_{v_2}[\chi_2((T_2)_w)]$ are connected and therefore $G_v[\chi(T_w)]$ would be connected. We assume that w is a minimal node such that $G_v[\chi(T_w)]$ is not connected, i.e., for every $u \in V(T_w) \setminus \{w\}$, $G_v[\chi(T_u)]$ is connected. Hence, there is $u \in \mathsf{Ch}_{T,r}(w)$ such that $\chi(w)$ is not connected to $G_v[\chi(T_u)]$. So $V(T_u) \cup V(T_i) = \emptyset$, since otherwise, the connectivity of $G_{v_i}[\chi_i((T_i)_w)]$ would imply the connectivity of $\chi(w)$ with $G_v[\chi(T_u)]$. Thus, there is an $x \in \mathsf{Leaf}(T,r)$ such that $u \in \mathsf{Anc}_{T,r}(x)$, and therefore $w \in \mathsf{Anc}_{T,r}(x)$, and $V(xTw) \cap \mathsf{Im}(f) = \emptyset$. So, according to item 5 of the $\diamondsuit_{\mathsf{join}}$ operation, there is $i \in \{1,2\}$ such that $xTw = xT_iw$. This contradicts the fact that $w \in V(T_i) \setminus \mathsf{Im}(f)$ and $V(T_u) \subseteq V(T_j) \setminus \mathsf{Im}(f)$ where $\{i,j\} = \{1,2\}$. Thus, (T, χ, r) is an $\mathcal{F}$-elimination tree of G_v. Therefore, (χ, σ) witnesses that $\hat{T} \in \mathcal{A}$.

Conversely, let $\hat{T} \in \mathcal{A}$ with witness pair (χ, σ). Let (χ_1, σ_1) and (χ_2, σ_2) be the co-restrictions of (χ, σ) to G_{v_1} and G_{v_2}, respectively.

Let T_1 be the tree obtained from T by removing the nodes not in $\mathsf{Im}(\chi_1)$ and adding edges between the parent and children of each removed node. Let r_1 be the root of T_1 and $h_1 := \mathsf{height}_{T_1,r_1}$. Let $\mathbf{R}_1$ be obtained from $\mathbf{R}$ by removing the vertices not in $\mathsf{Im}(\sigma_1)$. Then it is easy to see that $\hat{T}_1 = (T_1, r_1, h_1, \mathbf{R}_1, f)$ belongs to B_1 with witness pair (χ_1, σ_1). We construct similarly $\hat{T}_2 = (T_2, r_2, h_2, \mathbf{R}_2, f) \in B_2$ with witness pair (χ_2, σ_2).

Moreover, we claim that $\hat{T} \in \mathsf{join}^*(\hat{T}_1, \hat{T}_2)$. To prove this claim, the less trivial part is to show that T respects item 5 of the $\diamondsuit_{\mathsf{join}}$ operation. Let $x \in \mathsf{Leaf}(T,r)$ and $w \in \mathsf{Anc}_{T,r}(x)$ such that $V(xTw) \cap \mathsf{Im}(f) = \emptyset$. Suppose towards a contradiction that $V(xTw) \cap V(T_1) \neq \emptyset$ and $V(xTw) \cap V(T_2) \neq \emptyset$. Thus, there exist $u_1 \in V(xTw) \cap V(T_1) \setminus \mathsf{Im}(f)$ and $u_2 \in V(xTw) \cap V(T_2) \setminus \mathsf{Im}(f)$. Without loss of generality, suppose that $x \in V(T_1)$. We take such a node u_2 that is closest to x and u_1 to be the node just before u_2 in the path from x to u_2. Since $V(xTu_1) \subseteq V(T_1) \setminus \mathsf{Im}(f)$ and $u_2 \in V(T_2) \setminus \mathsf{Im}(f)$, $\chi(u_2)$ and $\chi(xTu_1)$ are not connected. This contradicts the fact that $G_v[\chi(T_{u_2})]$ is connected. Therefore, $\hat{T} \in \mathsf{join}^*(\hat{T}_1, \hat{T}_2)$, which implies that $\hat{T} \in \mathsf{join}^*(\mathcal{A}_1, \mathcal{A}_2)$. $\square$

Join procedure. Given two annotated trees $\hat{T}_1$ and $\hat{T}_2$, the *join procedure*, denoted by $\mathsf{join}(\hat{T}_1, \hat{T}_2)$, outputs $\mathsf{filter}_k \circ \mathsf{rep} \circ \mathsf{join}^*(\hat{T}_1, \hat{T}_2)$. See Figure 8.5 for an example. Note that we do not apply the crop operation since joining two cropped annotated trees gives a cropped annotated tree. Given two sets $\mathcal{A}_1$ and $\mathcal{A}_2$ of annotated trees, $\mathsf{join}(\mathcal{A}_1, \mathcal{A}_2)$ outputs $\bigcup_{\hat{T}_1 \in \mathcal{A}_1, \hat{T}_2 \in \mathcal{A}_2} \mathsf{join}(\hat{T}_1, \hat{T}_2)$.

Lemma 8.5.13. *Let G be a graph, (T, β, r) be a nice tree decomposition of G, v be a join node of T with children v_1 and v_2, and $\rho : \beta(v) \to [|\beta(v)|]$ be a bijection. Let k be an integer, $\mathcal{A} := \mathsf{char}_k(G_v, \beta(v), \rho)$, $\mathcal{A}_1 := \mathsf{char}_k(G_{v_1}, \beta(v_1), \rho)$, and $\mathcal{A}_2 := \mathsf{char}_k(G_{v_2}, \beta(v_2), \rho)$. Then $\mathcal{A} = \mathsf{join}_k(\mathcal{A}_1, \mathcal{A}_2)$.*

Proof. Let $\mathcal{B} := \mathsf{char}^*(G_v, \beta(v), \rho)$, $\mathcal{B}_1 := \mathsf{char}^*(G_{v_1}, \beta(v_1), \rho)$, and $\mathcal{B}_2 := \mathsf{char}^*(G_{v_2}, \beta(v_2), \rho)$. According to Lemma 8.5.12, $\mathcal{B} = \mathsf{join}^*(\mathcal{B}_1, \mathcal{B}_2)$.

Let $\hat{T} = (T, r, h, \mathbf{R}, f) \in \mathcal{B}$, $\hat{T}_1 = (T_1, r_1, h_1, \mathbf{R}_1, f_1) \in \mathcal{B}_1$, and $\hat{T}_2 = (T_2, r_2, h_2, \mathbf{R}_2, f_2) \in \mathcal{B}_2$, such that $\hat{T} \in \mathsf{join}^*(\hat{T}_1, \hat{T}_2)$. The complete join procedure joins $\mathbf{R}_1$ and $\mathbf{R}_2$ to obtain $\mathbf{R}$, without modifying their boundary nor deleting vertices or edges, so given two sets of annotated trees $\mathcal{C}_1$ and $\mathcal{C}_2$, $\mathsf{rep} \circ \mathsf{join}^*(\mathcal{C}_1, \mathcal{C}_2) = \mathsf{rep} \circ \mathsf{join}^*(\mathsf{rep}(\mathcal{C}_1), \mathsf{rep}(\mathcal{C}_2))$.

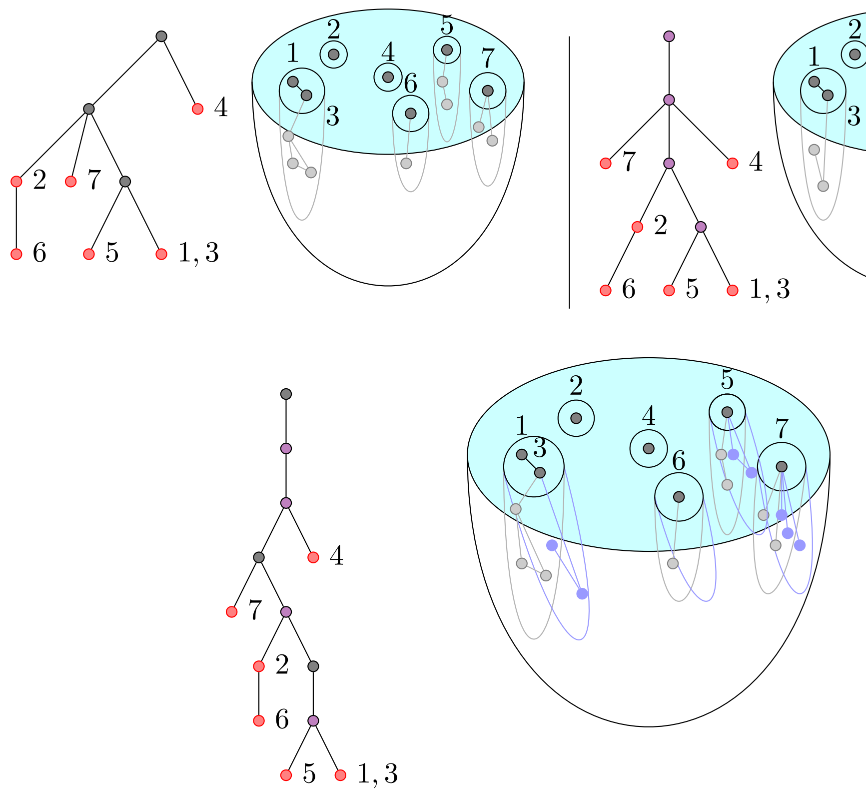

Figure 8.5: An annotated tree (below) obtained from the join procedure applied to the two annotated trees above.

The procedure can only increase the height of annotated trees, so $\mathsf{filter}_k \circ \mathsf{join}^*(\mathcal{C}_1, \mathcal{C}_2) = \mathsf{filter}_k \circ \mathsf{join}^*(\mathsf{filter}_k(\mathcal{C}_1), \mathsf{filter}_k(\mathcal{C}_2))$.

Moreover, item 6 of the $\diamondsuit_{\mathsf{join}}$ operation implies that, if for $w \in V(T)$, $V(T_w) \cap \mathsf{Im}(f) = \emptyset$, then there is $i \in \{1,2\}$ such that $T_w = (T_i)_w$. Therefore, each cropped subtree of T is exactly a cropped subtree of T_1 or a cropped subtree of T_2. Thus, $\mathsf{crop} \circ \mathsf{join}^*(\mathcal{C}_1, \mathcal{C}_2) = \mathsf{join}^*(\mathsf{crop}(\mathcal{C}_1), \mathsf{crop}(\mathcal{C}_2))$.

Since these three operations are commutative, we have $\mathcal{A} = \mathcal{M}_k(\mathcal{B}) = \mathsf{filter}_k \circ \mathsf{rep} \circ \mathsf{join}^*(\mathcal{M}_k(\mathcal{B}_1), \mathcal{M}_k(\mathcal{B}_2)) = \mathsf{join}_k(\mathcal{A}_1, \mathcal{A}_2)$. □

For the join procedure, we finally prove that it can generate a bounded number of annotated trees.

Lemma 8.5.14. *Let w and k be two positive integers, let $\mathbf{G}_1$ and $\mathbf{G}_2$ be two compatible w-boundaried graphs, let $\hat{T}_1 \in \mathsf{char}_k(\mathbf{G}_1)$, and let $\hat{T}_2 \in \mathsf{char}_k(\mathbf{G}_2)$. Then $|\mathsf{join}_k(\hat{T}_1, \hat{T}_2)| = 2^{\mathcal{O}(w \cdot k)}$.*

Proof. Let $\hat{T}_1 = (T_1, r_1, h_1, \mathbf{R}_1, f_1)$ and $\hat{T}_2 = (T_2, r_2, h_2, \mathbf{R}_2, f_2)$. We will show that $|(T_1, r_1, f_1) \diamondsuit_{\mathsf{join}} (T_2, r_2, f_2)| = 2^{\mathcal{O}(w \cdot k)}$. Let $(T, r, f) \in (T_1, r_1, f_1) \diamondsuit_{\mathsf{join}} (T_2, r_2, f_2)$. Notice that (T, r) has at most w leaves and height at most k. Each leaf of (T, r) is in $\mathsf{Im}(f) = \mathsf{Im}(f_1) = \mathsf{Im}(f_2)$, so it corresponds to both a leaf of (T_1, r_1) and a leaf of (T_2, r_2). Also note that T is obtained by choosing, for each path from a leaf v to r, a subset that corresponds to the path vT_1r (the rest is the path vT_2r). There are at most w such paths from a leaf to r, and each of them has length at most $k+1$. So $|(T_1, r_1, f_1) \diamondsuit_{\mathsf{join}} (T_2, r_2, f_2)| \leq 2^{w(k+1)}$. Then, to construct an annotated tree $(T, r, h, \mathbf{R}, f)$, the function h and the boundaried graph $\mathbf{R}$ are totally determined by T, h_1, h_2, $\mathbf{R}_1$, and $\mathbf{R}_2$. So we obtain the desired result. □

8.5.4 The algorithm

We finally present a recursive algorithm (Algorithm 1) that computes the elimination distance to $\mathsf{exc}(\mathcal{F})$ of a graph of bounded treewidth and proves Theorem 8.3.2. More precisely, given a boundary graph $\mathbf{G}$, a nice tree decomposition, and an integer k, the algorithm outputs $\mathsf{char}_k(\mathbf{G})$.

Algorithm 1: $\texttt{recEd}(\mathbf{G}, \mathcal{T}, k, v)$

Input: A connected boundaried graph $\mathbf{G} = (G, X, \rho)$, a nice tree decomposition $\mathcal{T} = (T, \beta, r)$ of G, an integer k, and a node $v \in V(T)$ such that $X = \beta(v)$.
Output: The characteristic $\mathsf{char}_k(\mathbf{G})$ of $\mathbf{G}$.

$\mathcal{A} \leftarrow \emptyset$
$w \leftarrow |\beta(v)|$
if *v is a leaf* **then**
  $\mathcal{A} \leftarrow \{\mathbb{1}\}$
end
else if *v is a forget node with child v' and forgotten vertex x* **then**
  $\rho' \leftarrow \rho \cup (x \mapsto w + 1)$
  $\mathcal{A}' \leftarrow \texttt{recEd}((G, \beta(v'), \rho'), \mathcal{T}, k, v')$
  $\mathcal{A} \leftarrow \mathsf{forget}(\mathcal{A}')$
end
else if *v is an introduce node with child v' and introduced vertex x* **then**
  $\tau \leftarrow (x \leftrightarrow \rho^{-1}(w))$
  $\rho' \leftarrow \rho \circ \tau$
  $\mathcal{A}' \leftarrow \texttt{recEd}((G, \beta(v'), \rho'|_{\beta(v')}), \mathcal{T}, k, v')$
  $N \leftarrow N_{G[\beta(v)]}(x)$
  $\mathcal{A} \leftarrow \tau(\mathsf{intr}_k(\mathcal{A}', \rho'(N)))$
end
else if *v is a join node with children v_1 and v_2* **then**
  $\mathcal{A}_1 \leftarrow \texttt{recEd}((G, \beta(v), \rho), \mathcal{T}, k, v_1)$
  $\mathcal{A}_2 \leftarrow \texttt{recEd}((G, \beta(v), \rho), \mathcal{T}, k, v_2)$
  $\mathcal{A} \leftarrow \mathsf{join}_k(\mathcal{A}_1, \mathcal{A}_2)$
end
return $\mathcal{A}$

Note that using backtracking in Algorithm 1, we can easily construct an annotated tree of minimum height in $\mathsf{char}^*(\mathbf{G})$ as well as its witness pair. In other words, given a connected graph G such that $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) \leq k$, we can construct an $\mathcal{F}$-elimination tree of G of height $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G)$ using Algorithm 1.

Lemma 8.5.15. *Given a connected graph G, an integer k, a nice tree decomposition $\mathcal{T} = (T, \beta, r)$ of G of width w, a bijection $\rho : \beta(r) \to [|\beta(r)|]$, and $v \in V(T)$, $\texttt{recEd}((G, \beta(v), \rho), \mathcal{T}, k, v)$ outputs $\mathsf{char}_k(G_v, \beta(v), \rho)$. Moreover, $\texttt{recEd}((G, \beta(r), \rho), \mathcal{T}, k, r)$ outputs $\mathsf{char}_k(G, \beta(r), \rho)$ in time $2^{\mathcal{O}_{\ell_{\mathcal{F}}}(w \cdot k + w \log w)} \cdot n$.*

Proof. We prove that for every $v \in V(T)$, by induction on the height of T_v, the algorithm $\texttt{recEd}$ of Algorithm 1 with input $((G, \beta(v), \rho), \mathcal{T}, k, v)$ returns $\mathsf{char}_k(G_v, \beta(v), \rho)$. Indeed, if v is a leaf, then $\mathsf{char}_k(G_v, \beta(v), \rho) = \{\hat{\mathbb{1}}\} = \texttt{recEd}((G, \beta(v), \rho), \mathcal{T}, k, v)$. Otherwise, v is either a forget node, or an introduce node, or a join node, and the correctness of the algorithm is implied from the induction hypothesis and Lemma 8.5.8, Lemma 8.5.10, and Lemma 8.5.13, respectively. Finally, since $G = G_r$, we have $\texttt{recEd}((G, \beta(r), \rho), \mathcal{T}, k, r) = \mathsf{char}_k(G, \beta(r), \rho)$.

We now analyze the running time. A nice tree decomposition of width w constructed by Proposition 4.3.3 has $\mathcal{O}(w \cdot n)$ bags, hence the linear dependence follows.

Let us first analyze the join procedure. During this procedure, we recursively obtain two characteristics of size $2^{\mathcal{O}_{\ell_\mathcal{F}}(w \cdot k + w \log w)}$ according to Lemma 8.5.6. Each pair of annotated trees can be joined in $2^{\mathcal{O}(w \cdot k)}$ ways, according to Lemma 8.5.14. Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be the boundaried graphs of such a pair of annotated trees. There is an integer $z \leq w$ such that they belong to $\mathcal{R}^z_{\ell_\mathcal{F}}$. By Proposition 4.4.1, $\mathbf{R}_1$ and $\mathbf{R}_2$ have size $\mathcal{O}_{\ell_\mathcal{F}}(z)$. So $\mathbf{R} := \mathbf{R}_1 \bigoplus \mathbf{R}_2$ has size $\mathcal{O}_{\ell_\mathcal{F}}(z)$ as well. The representation operation applied to $\mathbf{R}$ during the join operation for those two annotated trees finds the representative of $l \leq z$ boundaried graphs of respective boundaries of sizes $z_1, \ldots, z_l$ with $\sum_{i=1}^{l} z_i \leq z$ and with $\mathcal{O}_{\ell_\mathcal{F}}(z_i)$ vertices in the underlying graph due to Proposition 4.4.1. So by Lemma 4.4.3, the representation operation in the join procedure takes time $\sum_{i=1}^{l} 2^{\mathcal{O}_{\ell_\mathcal{F}}(w_i \log w_i)} = 2^{\mathcal{O}_{\ell_\mathcal{F}}(w \log w)}$. Checking that this is an annotated tree, that its height is at most d, and that we did not already create it also takes time $2^{\mathcal{O}_{\ell_\mathcal{F}}(w \cdot k + w \log w)}$. Hence, the total running time of the join procedure is $2^{\mathcal{O}_{\ell_\mathcal{F}}(w \cdot k + w \log w)}$.

It is easy to see that the forget procedure applied to an annotated tree creates at most one annotated tree. Moreover, the introduce procedure applied to an annotated tree creates $\mathcal{O}(w \cdot k)$ annotated trees according to Lemma 8.5.11. Similarly, the representation, crop, and filter operations in these procedures take time $2^{\mathcal{O}_{\ell_\mathcal{F}}(w \cdot k + w \log w)}$. So the lemma follows. □

We can finally prove Theorem 8.3.2.

Proof of Theorem 8.3.2. If $\mathcal{F} = \{K_1\}$, the algorithm of Proposition 8.5.4 outputs the desired result in time $2^{\mathcal{O}(\mathsf{tw} \cdot k)} \cdot n$. So let us assume that $\mathcal{F}$ is non-trivial.

Suppose first that G is connected. By Proposition 4.3.1 and Proposition 4.3.3, we can obtain a nice tree decomposition $\mathcal{T} = (T, \beta, r)$ of width $2\mathsf{tw} + 1$ in time $2^{\mathcal{O}(\mathsf{tw})} \cdot n$. Let $\rho : \beta(r) \to [|\beta(r)|]$ be an arbitrary ordering on the vertices of $\beta(r)$. We apply $\texttt{recEd}$ with input $((G, \beta(r), \rho), \mathcal{T}, k, r)$. By Lemma 8.5.15, this gives the desired result in time $2^{\mathcal{O}_{\ell_\mathcal{F}}(\mathsf{tw}(k + \log \mathsf{tw}))} \cdot n$.

If G is not connected, we apply the same procedure on each connected component. The running time is the same as in the above case. □

8.5.5 Exchangeability of boundaried graphs with the same characteristic

We give here a simple technical lemma on characteristics that will be used in Section 8.6. We show that boundaried graphs with the same characteristic can be exchanged, i.e., give graphs of the same elimination distance to $\mathcal{F}$ when "glued" to the same boundaried graph.

Given a positive integer k and a (possibly disconnected) boundaried graph $\mathbf{G}$, we define $\mathsf{char}_k(\mathbf{G})$ as $(\mathsf{char}_k(\mathbf{C}))_{\mathbf{C} \in \mathsf{cc}(\mathbf{G})}$. Note that we still have $|\mathsf{char}_k(\mathbf{G})| = 2^{\mathcal{O}_{\ell_\mathcal{F}}(w \cdot k + w \log w)}$. Therefore, we can extend $f_{8.5.6}$ so that $|\mathsf{char}_k(\mathbf{G})| \leq f_{8.5.6}(w, k)$ with $f_{8.5.6}(w, k) = 2^{\mathcal{O}_{\ell_\mathcal{F}}(w \cdot k + w \log w)}$ for any boundaried graph $\mathbf{G}$.

Lemma 8.5.16. *Let $\mathbf{G}$, $\mathbf{G}'$, and $\mathbf{G}''$ be three compatible boundaried graphs and let k be an integer such that $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\mathbf{G} \oplus \mathbf{G}'') \leq k$ and $\mathsf{char}_k(\mathbf{G}) = \mathsf{char}_k(\mathbf{G}')$. Then $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\mathbf{G} \oplus \mathbf{G}'') = \mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\mathbf{G}' \oplus \mathbf{G}'')$.*

Proof. We suppose without loss of generality that $\mathbf{G} \oplus \mathbf{G}''$, and therefore $\mathbf{G}' \oplus \mathbf{G}''$ as well, is connected. Indeed, if this is not the case, we may apply the following proof to each one of the connected components separately.

Let $\mathsf{cc}(\mathbf{G}) = \{\mathbf{C}_1, ..., \mathbf{C}_l\}$ and $\mathsf{cc}(\mathbf{G}') = \{\mathbf{C}'_1, ..., \mathbf{C}'_l\}$, such that $\mathsf{char}_k(\mathbf{C}_i) = \mathsf{char}_k(\mathbf{C}'_i)$ for $i \in [l]$. Let $i \in [l]$. We write $\mathbf{C}_i = (C_i, B_i, \rho_i)$. Let $\mathcal{T}_i = (T_i, \beta_i, r_i)$ be a nice tree decomposition of $\mathbf{C}_i$, i.e., such that $\beta_i(r_i) = B_i$. Since the B_i's are pairwise disjoint, there is a rooted tree decomposition $\mathcal{T}^* = (T^*, \beta^*, r)$ of G'', where G'' is the underlying graph of $\mathbf{G}''$, such that, for $i \in [l]$, there is $v_i \in \mathsf{Leaf}(T^*, r)$ with $\beta^*(v_i) = B_i$. Let $\mathcal{T} = (T, \beta, r)$ be the tree decomposition obtained from $\mathcal{T}^*$ and the $\mathcal{T}_i$'s by identifying v_i with r_i for $i \in [l]$ and adding nodes in $\mathcal{T}^*$ using Proposition 4.3.3, so that $\mathcal{T}$ is a nice tree decomposition of $\mathbf{G} \oplus \mathbf{G}''$.

Let $\mathcal{T}' = (T', \beta', r)$ be a nice tree decomposition of the graph $\mathbf{G}' \oplus \mathbf{G}''$ obtained from $\mathcal{T}$ by replacing $\mathcal{T}_i$ by a nice tree decomposition $\mathcal{T}'_i = (T'_i, \beta'_i, r'_i)$ of $\mathbf{C}'_i$ for $i \in [l]$. Observe that, for every $i \in [l]$, according to Lemma 8.5.15, $\mathtt{recEd}((\mathbf{G} \oplus \mathbf{G}'', \beta(v_i), \rho_{v_i}), \mathcal{T}, k, v_i) = \mathsf{char}_k(\mathbf{G} \oplus \mathbf{G}'', \beta(v_i), \rho) = \mathsf{char}_k(\mathbf{C}_i)$. Similarly, $\mathtt{recEd}((\mathbf{G}' \oplus \mathbf{G}'', \beta'(v_i), \rho_{v_i}), \mathcal{T}', k, v_i) = \mathsf{char}_k(\mathbf{C}'_i)$. Thus, Algorithm 1 applied with input $((\mathbf{G} \oplus \mathbf{G}'', \beta(v_i), \rho_{v_i}), \mathcal{T}, k, v_i)$ and $((\mathbf{G}' \oplus \mathbf{G}'', \beta'(v_i), \rho_{v_i}), \mathcal{T}', k, v_i)$ outputs the same result.

We next set $U := V(T) \setminus \bigcup_{i \in [l]} V(T_i)$. Note that $U = V(T') \setminus \bigcup_{i \in [l]} V(T'_i)$. Therefore, in each node u of U, Algorithm 1 applied with input $((\mathbf{G} \oplus \mathbf{G}'', \beta(u), \rho_u), \mathcal{T}, k, u)$ and $((\mathbf{G}' \oplus \mathbf{G}'', \beta'(u), \rho_u), \mathcal{T}', k, u)$ outputs the same result. Thus, Algorithm 1 applied with input $((\mathbf{G} \oplus \mathbf{G}'', \beta(r), \rho_r), \mathcal{T}, k, r)$ and $((\mathbf{G}' \oplus \mathbf{G}'', \beta'(r), \rho_r), \mathcal{T}', k, r)$ outputs the same result. So $\mathsf{char}_k(\mathbf{G} \oplus \mathbf{G}'', \beta(r), \rho_r) = \mathsf{char}_k(\mathbf{G}' \oplus \mathbf{G}'', \beta'(r), \rho_r)$. Therefore, according to Lemma 8.5.5, $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\mathbf{G} \oplus \mathbf{G}'') = \mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\mathbf{G}' \oplus \mathbf{G}'')$. $\square$

8.6 Bounding the obstructions of $\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$

In this section, we prove the following result that provides an upper bound on the size of the graphs in $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$. The following theorem is a reformulation of Theorem 2.4.4.

Theorem 8.6.1. *Let $\mathcal{F}$ be a non-empty finite collection of non-empty graphs and k be a positive integer. Every graph in $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$ has $2^{2^{2^{2^{k^{\mathcal{O}_{\ell_\mathcal{F}}(1)}}}}}$ vertices. In the particular case when $\mathcal{F}$ contains an apex-graph, every graph in $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$ has $2^{2^{k^{\mathcal{O}_{\ell_\mathcal{F}}(1)}}}$ vertices.*

Recall that when $\mathcal{F} = \{K_1\} = \mathsf{obs}(\mathcal{H}_\emptyset)$, it is known [100] that every graph in $\mathsf{obs}(\mathcal{E}_k(\mathcal{H}_\emptyset))$ has at most $2^{2^{k-1}}$ vertices.

Theorem 8.6.1 implies that one can construct an algorithm that receives as input $\mathsf{obs}(\mathsf{exc}(\mathcal{F}))$ and k, and outputs $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$. This is done by enumerating all graphs on at most $f(k)$ vertices, where $f(k)$ is the bound on the number of vertices given by Theorem 8.6.1, and filtering out those that are members of $\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$ and taking those that are minor-minimal in what is left. The running time of the algorithm can be bounded by $\mathcal{O}(2^{2^{2^{2^{2^{k^{\mathcal{O}_{\ell_\mathcal{F}}(1)}}}}}} \cdot n^2)$ in the general case and by $\mathcal{O}(2^{2^{2^{k^{\mathcal{O}_{\ell_\mathcal{F}}(1)}}}} \cdot n^2)$ if $\mathcal{F}$ contains an apex-graph.

Note that this brute-force algorithm can be used to solve Elimination Distance to $\mathsf{exc}(\mathcal{F})$. Indeed, to solve Elimination Distance to $\mathsf{exc}(\mathcal{F})$, we can compute $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$ and then check whether there is a graph in $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$ that is a minor of the input graph. Of course, this algorithm is much less efficient than the ones presented in the previous sections.

The rest of the section is structured as follows: in Subsection 8.6.1 we bound the treewidth of a minor-minimal obstruction of $\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$, while in Subsection 8.6.2 we bound the size of a minor-minimal obstruction of $\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$ of small treewidth. This immediately implies Theorem 8.6.1.

8.6.1 Bounding the treewidth of an obstruction

In this subsection we aim to prove an upper bound on the treewidth of a minor-minimal obstruction of $\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$.

Lemma 8.6.2. *Let $\mathcal{F}$ be a finite collection of graphs. There exists a function $f_{8.6.2} : \mathbb{N}^3 \to \mathbb{N}$ such that if $G \in \mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$, then $\mathsf{tw}(G) \leq f_{8.6.2}(k, a_{\mathcal{F}}, s_{\mathcal{F}})$. Moreover, $f_{8.6.2}(k, a, s) = 2^{\log(k\cdot c)\cdot 2^{k^{a-1}\cdot 2^{\mathcal{O}(s^2 \log s)}}}$, where $a = a_{\mathcal{F}}$, $s = s_{\mathcal{F}}$, and c is a constant depending on $\ell_{\mathcal{F}}$.*

Note that when $a_{\mathcal{F}} = 1$, $f_{8.6.2}(k, 1, s) = \mathcal{O}_{\ell_{\mathcal{F}}}(k^{2^{2^{\mathcal{O}(s^2 \log s)}}})$.

Proof. For simplicity, we use s, a, and ℓ instead of $s_{\mathcal{F}}, a_{\mathcal{F}}$, and $\ell_{\mathcal{F}}$, respectively. We set

$$\begin{aligned} b &= g_{4.6.2}(s) + k + 1, & d &= g_{8.2.4}(a-1, \ell), \\ r_4 &= f_{8.2.4}(a-1, \ell, 3, k(k+1)/2), & r_3 &= f_{4.6.12}(r_4, a-1, b, d), \\ x &= g_{8.2.5}(a, s, k(k+1)/2), & p &= h_{8.2.5}(a, s, k(k+1)/2), \\ l &= (x-1)\cdot b, & r_2 &= \mathsf{odd}(\max\{f_{4.6.8}(l+1, r_3, p), f_{8.2.5}(a, s, k(k+1)/2)\}), \\ r_1 &= f_{4.6.2}(s)\cdot r_2, \text{ and} & w &= f_{4.6.1}(s)\cdot r_1 + k + 1. \end{aligned}$$

It is easy to verify that $w = 2^{\log(k\cdot c)\cdot 2^{k^{a-1}\cdot 2^{\mathcal{O}(s^2 \log s)}}}$.

Suppose towards a contradiction that $\mathsf{tw}(G) > w$. Since $G \in \mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$, for each $v \in V(G)$, $G - v \in \mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$. Therefore, there exists a $(k+1)$-elimination set S of G for $\mathsf{exc}(\mathcal{F})$. Thus, for each $C \in \mathsf{cc}(G - S)$, $C \in \mathsf{exc}(\mathcal{F})$. According to Lemma 8.2.3, $\mathsf{tw}(G - S) > w - k - 1 = f_{4.6.1}(s)\cdot r_1$, so there is $C \in \mathsf{cc}(G - S)$, such that $\mathsf{tw}(C) > f_{4.6.1}(s)\cdot r_1$. Moreover, K_s is not a minor of C. Therefore, according to Proposition 4.6.1, C contains an r_1-wall W_1.

Since $r_1 = f_{4.6.2}(s)\cdot r_2$, by Proposition 4.6.2, there is a set $A \subseteq V(C)$ of size at most $g_{4.6.2}(s)$ and a flatness pair $(W_2, \mathfrak{R}_2)$ of $C - A$ of height r_2 such that W_2 is a tilt of a subwall of W_1. Due to Proposition 4.6.7, there is a regular flatness pair $(W_2', \mathfrak{R}_2')$ of $C - A$ of height r_2.

Since S is a $(k+1)$-elimination set of G for $\mathsf{exc}(\mathcal{F})$ and $C \in \mathsf{cc}(G - S)$, there exists a set $P \subseteq S$ of size at most $k+1$ such that $(L, R) := (V(G) \setminus V(C), V(C) \cup P)$ is a separation of G with $L \cap R = P$. Thus, if $\mathfrak{R}_2''$ is the 5-tuple obtained by adding the vertices of $G - (C \cup P)$ to the set in the first coordinate of $\mathfrak{R}_2'$, $(W_2', \mathfrak{R}_2'')$ is a regular flatness pair of $G - (A \cup P)$ of height r_2.

Let $\tilde{\mathcal{Q}}$ be a W_2'-canonical partition of $G - (A \cup P)$. Let B be the set of vertices of $A \cup P$ adjacent to vertices of at least x p-internal bags of $\tilde{\mathcal{Q}}$. Let $\mathcal{W} = \{W^1, \ldots, W^{l+1}\}$ be a family of $l+1$ r_3-subwalls of W_2' such that for every $i \in [l+1]$, $\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^i)$ is a subgraph of $\bigcup\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{\mathcal{Q}}\}$ and for every $i, j \in [l+1]$ with $i \neq j$, there is no internal bag $Q \in \tilde{\mathcal{Q}}$ that contains vertices of both $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^i))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^j))$. The existence of $\mathcal{W}$ follows from the fact that $r_2 \geq f_{4.6.8}(l+1, r_3, p)$ and Proposition 4.6.8. Notice that the set $N_G((A \cup P) \setminus B)$ intersects the vertex set of at most $(x-1)\cdot|(A \cup P) \setminus B| \leq l$ p-internal bags of $\tilde{\mathcal{Q}}$. Thus, there is an $i \in [l+1]$ such that no vertex in $(A \cup P) \setminus B$ is adjacent to vertices of $\bigcup \mathsf{influence}_{\mathfrak{R}_2''}(W^i)$.

Let $(W_3, \mathfrak{R}_3)$ be a W^i-tilt of $(W_2', \mathfrak{R}_2'')$. Since $|B| \leq |A \cup P| \leq g_{4.6.2}(s) + k + 1 = b$ and $r_3 = f_{4.6.12}(r_4, b, a-1, d)$, by Proposition 4.6.12, there is a flatness pair $(W_4, \mathfrak{R}_4)$ of $G - B$ of height r_4 that is d-homogeneous with respect to $\binom{B}{<a}$ and is a tilt of a subwall of $(W_3, \mathfrak{R}_3)$. By Observation 4.6.4 and Observation 4.6.5, $(W_4, \mathfrak{R}_4)$ is regular.

Recall that $(W_3, \mathfrak{R}_3)$ is a W^i-tilt of $(W_2', \mathfrak{R}_2'')$, $(W_4, \mathfrak{R}_4)$ is a tilt of a subwall of $(W_3, \mathfrak{R}_3)$, and $(W_4, \mathfrak{R}_4)$ is a flatness pair of $G - B$ with $B \subseteq A \cup P$. Thus, given a W_4-canonical partition $\tilde{\mathcal{Q}}_1$ of $G - B$, there is a W_2'-canonical partition $\tilde{\mathcal{Q}}_2$ of $G-(A\cup P)$ such that each internal bag of $\tilde{\mathcal{Q}}_1$ is contained in an internal bag of $\tilde{\mathcal{Q}}_2$. Therefore, for every set $X\subseteq V(G)$, $\mathsf{bid}_{G-B,W_4}(X)\leq \mathsf{bid}_{G-(A\cup P),W_2'}(X)$.

Moreover, since $r_2\geq f_{8.2.5}(a,s,k(k+1)/2)$, according to Proposition 8.2.5, every subset of B of size a intersects every set $X\subseteq V(G)$ such that $G-X\in\mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-(A\cup P),W_2'}(X)\leq k(k+1)/2$. Hence, for any such X, $|B\setminus X|<a$.

Thus, according to Proposition 8.2.4, since $r_4=f_{8.2.4}(a-1,\ell,3,k(k+1)/2)$, it holds that there is a vertex v such that, for every set $X\subseteq V(G)$ with $G-X\in\mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-B,W_4}(X)\leq k(k+1)/2$, $G-X\in\mathsf{exc}(\mathcal{F})$ if and only if $G-(X\setminus v)\in\mathsf{exc}(\mathcal{F})$.

Lemma 8.2.7 implies that for any k-elimination set S' of $G-v$ for $\mathsf{exc}(\mathcal{F})$, there is a set $X\supseteq S'$ such that $G-X\in\mathsf{exc}(\mathcal{F})$ and $\mathsf{bid}_{G-(A\cup P),W_2'}(X)\leq k(k+1)/2$. Since $\mathsf{bid}_{G-B,W_4}(X)\leq\mathsf{bid}_{G-(A\cup P),W_2'}(X)$, we also have that $\mathsf{bid}_{G-B,W_4}(X)\leq k(k+1)/2$. Thus, $G-X\in\mathsf{exc}(\mathcal{F})$ if and only if $G-(X\setminus v)\in\mathsf{exc}(\mathcal{F})$ and $G\in\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$ if and only if $G-v\in\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$. However, since $G\in\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$, it holds that $G\notin\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$ and $G-v\in\mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$, a contradiction. □

8.6.2 Bounding the size of an obstruction of small treewidth

In order to bound the size of an obstruction of small treewidth, we first present some additional notions on tree decompositions on boundaried graphs.

Treewidth of boundaried graphs. Let $\mathbf{G}=(G,B,\rho)$ be a boundaried graph. A *tree decomposition* of $\mathbf{G}$ is a rooted tree decomposition (T,β,r) of G such that $\beta(r)=B$. The *width* of (T,β,r) is the width of (T,β). The treewidth of a boundaried graph $\mathbf{G}$ is the minimum width over all its tree decompositions and is denoted by $\mathsf{tw}(\mathbf{G})$. A *nice tree decomposition* of $\mathbf{G}$ is a tree decomposition (T,β,r) of $\mathbf{G}$ that is also a nice tree decomposition of G rooted at r.

Let $\mathbf{G}=(G,B,\rho)$ be a boundaried graph and $\mathcal{T}=(T,\beta,r)$ be a tree decomposition of $\mathbf{G}$. Notice that if $a,b\in V(T)$ and $a\in\mathsf{Anc}_{T,r}(b)$, then G_b is a subgraph of G_a. We define the t_q-boundaried graph $\bar{\mathbf{G}}_q=(\bar{G}_q,\beta(q),\rho_q)$, where $\bar{G}_q=G-(V(G_q)\setminus\beta(q))$. Notice that $\mathbf{G}_q$ and $\bar{\mathbf{G}}_q$ are compatible and $\mathbf{G}_q\oplus\bar{\mathbf{G}}_q=G$.

Linked tree decompositions. Our next step is to use a special type of tree decompositions, namely *linked tree decompositions*, defined by Robertson and Seymour in [261]. Thomas in [304] proved that every graph G admits a linked tree decomposition of width $\mathsf{tw}(G)$ (see also [28, 106]). By combining the result of [304] and [55, Lemma 4], we can consider tree decompositions as asserted in the following result.

Proposition 8.6.3 ([55]). *Let $t\in\mathbb{N}_{\geq 1}$. For every boundaried graph $\mathbf{G}=(G,B,\rho)$ of treewidth $t-1$, there exists a tree decomposition (T,β,r) of $\mathbf{G}$ of width $t-1$ such that*

1. *(T,r) is a binary tree,*
2. *for every $a,b\in V(T)$ where a is a child of b in (T,r), if $|\beta(a)|=|\beta(b)|$ then G_a is a proper subgraph of G_b, i.e., $|V(G_a)|<|V(G_b)|$,*
3. *for every $s\in\mathbb{N}$ and every pair $u_1,u_2\in V(T)$, where $u_1\in\mathsf{Anc}_{T,r}(u_2)$ and $|\beta(u_1)|=|\beta(u_2)|$, either there is an internal vertex w of u_1Tu_2 such that $|\beta(w)|<s$, or there exists a collection of s vertex-disjoint paths in G between $\beta(u_1)$ and $\beta(u_2)$, and*
4. *$|V(G)|\leq t\cdot|V(T)|$.*

In fact, linked tree decompositions are defined as the tree decompositions satisfying only property (3) [261, 304]. In our proofs, we will need the extra properties (1), (2), and (4) that are provided by [55, Lemma 4].

We bound the size of a minor-minimal obstruction of small treewidth in Lemma 8.6.5. To do so, we need the following result (for a proof see e.g. [139, Lemma 14]).

Proposition 8.6.4 ([139]). *Let $r, m \in \mathbb{N}_{\geq 1}$ and w be a word of length m^r over the alphabet $[r]$. Then there is a number $k \in [r]$ and a subword u of w such that u contains only numbers not smaller than k and u contains the number k at least m times.*

We are now ready to prove Lemma 8.6.5. The idea is to apply the technique of Lagergren [213] combined with the bound on the number of characteristics provided in Subsection 8.5.5. The proof of Lemma 8.6.5 is very similar to the corresponding proof in [285] for $\mathsf{obs}(\mathcal{A}_k(\mathsf{exc}(\mathcal{F})))$.

Lemma 8.6.5. *Let $\mathcal{F}$ be a finite non-trivial collection of graphs. There exists a function $f_{8.6.5} : \mathbb{N}^2 \to \mathbb{N}$ such that if k is an integer and G is a graph in $\mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$ of treewidth tw, then $|V(G)| \leq f_{8.6.5}(\mathsf{tw}, k)$. Moreover, $f_{8.6.5}(t, k) = 2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(t^3 + k \cdot t^2)}}$.*

Proof. Let $G \in \mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$. We set $t := \mathsf{tw}(G) + 1$. For simplicity, we use ℓ instead of $\ell_{\mathcal{F}}$. We set

$$d = f_{8.5.6}(t-1, k) + 1, \qquad m = 2^{\binom{t}{2}} \cdot (d-1) + 1,$$
$$x = m^t, \text{ and} \qquad b = t \cdot 2^x.$$

It is easy to verify that $b = 2^{2^{\mathcal{O}_{\ell_{\mathcal{F}}}(t^3 + k \cdot t^2)}}$.

Suppose towards a contradiction that $|V(G)| > b$. Let (T, β) be a tree decomposition of G of width $\mathsf{tw}(G)$ and let $r \in V(T)$. We consider the rooted tree (T, r) and we set $B := \beta(r)$ and a bijection $\rho : B \to [|B|]$. We set $\mathbf{G} = (G, B, \rho)$ and observe that (T, β, r) is a tree decomposition of $\mathbf{G}$ of width $\mathsf{tw}(G)$. Since $\mathsf{tw}(\mathbf{G}) = \mathsf{tw}(G) = t - 1$, by Proposition 8.6.3, we can assume that for the tree decomposition (T, β, r) of $\mathbf{G}$ of width $t - 1$, Properties (1) to (4) are satisfied.

Since $|V(G)| > b = t \cdot 2^x$, Property (4) implies that $|V(T)| > 2^x$. Also, by Property (1), (T, r) is a binary tree and therefore there exists a leaf u of T such that $|V(rTu)| \geq x$. We set $l := |V(rTu)|$.

We set $v_1 = r$ and for every $i \in [l-1]$, we set v_{i+1} to be the child of v_i in (T, r) that belongs to $V(rTu)$. Keep in mind that $v_l = u$. For every $i \in [l]$, we set $c_i := |\beta(v_i)|$ and observe that, since (T, β, r) has width $t - 1$, $c_i \in [t]$.

Let C be the word $c_1 \cdots c_x$. Since $x = m^t$ and every $c_i \in [t]$, then, due to Proposition 8.6.4, there is a $t' \in [t]$ and a subword C' of C such that, for every c in C', $c \geq t'$ and there are at least m numbers in C' that are equal to t'. Therefore, there exists a set $\{z_1, \ldots, z_m\} \subseteq V(T)$ such that for every $i \in [2, m]$, z_i is a descendant of z_{i-1} in (T, r), for every $z' \in V(z_1 T z_m)$ it holds that $|\beta(z')| \geq t'$, and, for every $i \in [m]$, $|\beta(z_i)| = t'$. Hence, Property (3) of the tree decomposition (T, β, r) of $\mathbf{G}$ implies that there exists a collection $\mathcal{P} = \{P_1, \ldots, P_{t'}\}$ of t' vertex-disjoint paths in G between $\beta(z_1)$ and $\beta(z_m)$.

For every $i \in [m]$, let ρ_i be the function mapping a vertex v in $\beta(z_i)$ to the index of the path of $\mathcal{P}$ it intersects, i.e., for every $j \in [t']$, if v is a vertex in $V(P_j) \cap \beta(z_i)$, where $P_j \in \mathcal{P}$, then $\rho_i(v) = j$. Also, for every $i \in [m]$, let $\mathbf{G}_{z_i}$ be the t'-boundaried graph $(G_{z_i}, \beta(z_i), \rho_i)$. Since, $m = 2^{\binom{t}{2}} \cdot (d-1) + 1$, there is a set $J \subseteq [m]$ of size d such that for every $i, j \in J$, the graph $G_{z_i}[\beta(z_i)]$ is isomorphic to the graph $G_{z_j}[\beta(z_j)]$. Therefore, for every $i, j \in J$, $\mathbf{G}_{z_i}$ and $\mathbf{G}_{z_j}$ are compatible. Furthermore, observe that for every $i, j \in J$ with $i \leq j$, $\mathbf{G}_{z_j} \preceq_{\mathsf{m}} \mathbf{G}_{z_i}$. To see why this holds, for every $i, j \in J$ with $i < j$, let $\mathcal{P}_{i,j}$ be the collection of subpaths of $\mathcal{P}$ between the vertices of $\beta(z_i)$ and $\beta(z_j)$ and consider the graph $G_{z_i}[\beta(z_i)] \cup \bigcup \mathcal{P}_{i,j} \cup G_{z_j}$, which is a subgraph of G_{z_i}. By contracting the edges in $\mathcal{P}_{i,j}$, we obtain a boundaried graph isomorphic to $\mathbf{G}_{z_j}$.

Recall that $|J| = d = f_{8.5.6}(t-1, k) + 1$. Thus, by Lemma 8.5.6, there exist $i, j \in J$ such that j is the smallest element in J that is greater than i and $\mathsf{char}_k(\mathbf{G}_{z_i}) = \mathsf{char}_k(\mathbf{G}_{z_j})$. For simplicity, we set $a := z_i$ and $b := z_j$. Notice that, in G, by contracting the edges of the paths in $\mathcal{P}_{i,j}$ and removing the vertices of G_a that are not vertices of G_b, we obtain a graph isomorphic to $\bar{\mathbf{G}}_a \oplus \mathbf{G}_b$. Therefore, $\bar{\mathbf{G}}_a \oplus \mathbf{G}_b$ is a minor of G. Furthermore, $|V(\bar{\mathbf{G}}_a \oplus \mathbf{G}_b)| < |V(G)|$. To prove this, we argue that G_b is a proper subgraph of G_a. First recall that for every $y \in V(aTb)$, $|\beta(y)| \geq t'$. If there is a $y \in V(aTb)$ such that $|\beta(y)| > t'$, then there is a vertex $v \in \beta(y)$ that is a vertex of $V(G_a) \setminus V(G_b)$ and thus G_b is a proper subgraph of G_a, while in the case where for every $y \in V(aTb)$, $|\beta(y)| = t'$, Property (2) of Proposition 8.6.3 implies that G_b is a proper subgraph of G_a.

Let $G' := \bar{\mathbf{G}}_a \oplus \mathbf{G}_b$. Since $|V(G')| < |V(G)|$, G' is a minor of G, and $G \in \mathsf{obs}(\mathcal{E}_k(\mathsf{exc}(\mathcal{F})))$, it holds that $G' \in \mathcal{E}_k(\mathsf{exc}(\mathcal{F}))$. By Lemma 8.5.16, since $\mathsf{char}_k(\mathbf{G}_a) = \mathsf{char}_k(\mathbf{G}_b)$, we have that $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\bar{\mathbf{G}}_a \oplus \mathbf{G}_a) = \mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(\bar{\mathbf{G}}_a \oplus \mathbf{G}_b)$ and therefore $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) = \mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G')$. This contradicts the fact that $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G) > k$ and $\mathsf{ed}_{\mathsf{exc}(\mathcal{F})}(G') \leq k$. $\square$

Part IV

Towards generalization

CHAPTER 9

Dynamic programming for bipartite treewidth

Contents

In this chapter, we prove the results presented in Section 2.5, which are restated here for convenience.

Theorem 2.5.1. *Let $\mathcal{H}_t$ be the class of graphs that exclude K_t as a subgraph. Then, there is an algorithm that, given a graph G and a bipartite tree decomposition of G of width w, solves* Vertex Deletion to $\mathcal{H}_t$ *in time $\mathcal{O}(2^w \cdot (w^t \cdot (n+m) + m\sqrt{n}))$.*

Theorem 2.5.2. *There is an algorithm that, given a graph G and a bipartite tree decomposition of G of width w, solves* Weighted (resp. Unweighted) Vertex Cover *in time $\mathcal{O}(2^w \cdot (w \cdot (w+m) + m \cdot n))$ (resp. $\mathcal{O}(2^w \cdot (w \cdot (w+m) + m\sqrt{n}))$).*

Theorem 2.5.3. *There is an algorithm that, given a graph G and a bipartite tree decomposition of G of width w, solves* Odd Cycle Transversal *in time $\mathcal{O}(3^w \cdot w \cdot (m + k^2) \cdot n)$.*

In Table 9.1, we present all the results that we actually found concerning the parameterization by bipartite treewidth, though we only present the results on graph modification problems in this thesis. In Section 9.1, we give an overview of our dynamic programming scheme. In Section 9.2, we give several equivalent definitions of odd-minors. In Section 9.3, we define formally bipartite treewidth and the more general parameter of 1-$\mathcal{H}$-treewidth and give some preliminary results. In Section 9.4, we provide our dynamic programming scheme for solving problems parameterized by bipartite treewidth. Finally, in Section 9.5, we give applications of our scheme and discuss hardness results.

Problem	Complexity	Constraints on H/Running time
H-Minor-Cover [314] H(-Induced)-Subgraph/ Odd-Minor-Cover [314] and H(-Induced)-Subgraph/ H(-Odd)-Minor-Packing H-Scattered-Packing	para-NP-complete, $k = 0$	H containing P_3 as a subgraph H bipartite containing P_3 as a subgraph H 2-connected bipartite with $\|V(H)\| \geq 2$
3-Coloring	para-NP-complete, $k = 3$	
K_t-Subgraph-Cover Independent Set Weighted Independent Set Odd Cycle Transversal Maximum Weighted Cut	FPT	$\mathcal{O}(2^k \cdot (k^t \cdot (n+m) + m\sqrt{n}))$ $\mathcal{O}(2^k \cdot (k \cdot (k+n) + m\sqrt{n}))$ $\mathcal{O}(2^k \cdot (k \cdot (k+n) + n \cdot m))$ $\mathcal{O}(3^k \cdot k \cdot n \cdot (m+k^2))$ $\mathcal{O}(2^k \cdot (k \cdot (k+n) + n^{\mathcal{O}(1)}))$
H-Subgraph-Packing H-Induced-Subgraph-Packing H-Scattered-Packing H-Odd-Minor-Packing	XP	H non-bipartite 2-connected $n^{\mathcal{O}(k)}$

Table 9.1: Summary of the results obtained for problems parameterized by the bipartite treewidth k of the input graph.

9.1 Overview of the dynamic programming scheme

Compared to dynamic programming on classical tree decompositions, there are two main difficulties for doing dynamic programming on (rooted) *bipartite* tree decompositions. The first one is that the bags in a bipartite tree decomposition may be arbitrarily large, which prevents us from applying typical brute-force approaches to define table entries. The second one, and apparently more important, is the lack of an upper bound on the number of children of each node of the decomposition. Indeed, unfortunately, a notion of "nice bipartite tree decomposition" preserving the width (even approximately) does not exist (cf. Lemma 9.3.5).

For particular problems, however, we can devise ad-hoc solutions. Namely, for K_t-Subgraph-Cover, Weighted Vertex Cover/Independent Set, Odd Cycle Transversal, and Maximum Weighted Cut parameterized by btw, we overcome the above issue by managing to replace the children with constant-sized bipartite gadgets. More specifically, we guess an annotation of the "apex" vertices of each bag t, whose number is bounded by btw, that essentially tells which of these vertices go to the solution or not (with some extra information depending on each particular problem; for instance, for Odd Cycle Transversal, we also guess the side of the bipartition of the non-solution vertices). Having this annotation, each adhesion of the considered node t with a child contains, by the definition of bipartite tree decompositions, at most one vertex v that is not annotated. At this point, we crucially observe that, for the considered problems, we can make a local computation for each child, independent from the computations at other children, depending only on the values of the optimum solutions at that child that are required to *contain* or to *exclude* v (note that we need to be able to keep this extra information at the tables of the children). Using the information given by these local computations, we can replace the children of t by constant-sized bipartite gadgets (sometimes empty) so that the newly built graph, which we call a *nice reduction*, is an equivalent instance modulo some constant. If a nice reduction can be efficiently computed for a problem Π, then we say that Π is a *nice problem* (cf. Subsection 9.4.2). The newly modified bag has bounded oct, so we can then use an FPT-algorithm parameterized by oct to find the optimal solution with respect to the guessed annotation.

An illustrative example. Before entering into some more technical details and general definitions, let us illustrate this idea with the WEIGHTED VERTEX COVER problem. The formal definition of bipartite tree decomposition can be found in Section 9.2 (in fact, for the more general setting of 1-$\mathcal{H}$-treewidth). For this informal explanation, it is enough to suppose that we want to compute the dynamic programming tables at a bag associated with a node t of the rooted tree decomposition, and that the vertices of the bag at t are partitioned into two sets: $\beta(t)$ induces a bipartite graph and its complement, denoted by $\alpha(t)$, corresponds to the apex vertices, whose size is bounded by the parameter btw. The first step is to guess, in time at most 2^{btw}, which vertices in $\alpha(t)$ belong to the desired minimum vertex cover. After such a guess, all the vertices in $\alpha(t)$ can be removed from the graph, by also removing the neighborhood of those that were *not* taken into the solution. The definition of bipartite tree decomposition implies that, in each adhesion with a child of the current bag, there is at most one "surviving" vertex. Let v be such a vertex belonging to the adhesion with a child t' of t. Suppose that, inductively, we have computed in the tables for t' the following two values, subject to the choice that we made for $\alpha(t)$: the minimum weight w_v of a vertex cover in the graph below t' that contains v, and the minimum weight $w_{\bar{v}}$ of a vertex cover in the graph below t' that does *not* contain v. Then, the trick is to observe that, having these two values at hand, we can totally forget the graph below t': it is enough to delete this whole graph, except for v, and attach a new pendant edge vu, where u is a new vertex, such that v is given weight w_v and u is given weight $w_{\bar{v}}$. It is easy to verify that this gadget mimics, with respect to the current bag, the behavior of including vertex v or not in the solution for the child t'. Adding this gadget for every child results in a bipartite graph, for which we can just solve WEIGHTED VERTEX COVER in polynomial time using a classical algorithm [198, 246], and add the returned weight to our tables. The running time of this whole procedure, from the leaves to the root of the decomposition, is clearly FPT parameterized by the bipartite treewidth of the input graph.

Extensions and limitations. The algorithm sketched above is problem-dependent, in particular the choice of the gadgets for the children, and the deletion of the neighborhood of the vertices chosen in the solution. Which type of replacements and reductions can be afforded in order to obtain an FPT-algorithm for bipartite treewidth? For instance, concerning the gadgets for the children, as far as the considered problem can be solved in polynomial time on bipartite graphs, we could attach to the "surviving" vertices an arbitrary bipartite graph instead of just an edge. If we assume that the considered problem is FPT parameterized by oct (which is a reasonable assumption, as btw generalizes oct), then one could think that it may be sufficient to devise gadgets with bounded oct. Unfortunately, this will not work in general: even if each of the gadgets has bounded oct (take, for instance, a triangle), since we do not have any upper bound, in terms of btw, on the number of children (hence, the number of different adhesions), the resulting graph after the gadget replacement may have unbounded oct. In order to formalize the type of replacements and reductions that can be allowed, we introduce in Section 9.4 the notions of *nice reduction* and *nice problem*. Additional insights into these definitions, which are quite lengthy, are provided in Subsection 9.4.2.

Another sensitive issue is that of "guessing the vertices into the solution". While this is quite simple for Weighted Vertex Cover (either a vertex is in the solution, or it is not), for some other problems we may have to guess a richer structure in order to have enough information to combine the tables of the children into the tables of the current bag. This is the reason why, in the general dynamic programming scheme that we present in Section 9.4, we deal with *annotated problems*, i.e., problems that receive as input, apart from a graph, a collection of annotated sets in the form of a partition $\mathcal{X}$ of some $X \subseteq V(G)$. For instance, for Weighted Vertex Cover, we define its *annotated extension*, which we call Annotated Weighted Vertex Cover, that takes as input a graph G and two disjoint sets R and S of vertices of G, and asks for a minimum vertex cover $S^\star$ such that $S \subseteq S^\star$ and $S^\star \cap R = \emptyset$.

General dynamic programming scheme. Our general scheme essentially says that if a problem Π has an annotated extension Π' that is

- a nice problem and
- solvable in FPT-time parameterized by oct,

then Π is solvable in FPT-time parameterized by btw. More specifically, it is enough to prove that Π' is solvable in time $f(|X|) \cdot n^{\mathcal{O}(1)}$ on an instance $(G, \mathcal{X})$ such that $G - X$ is bipartite, where $\mathcal{X}$ is a partition of X corresponding to the annotation. This general dynamic programming algorithm works in a wider setting, namely for a general graph class $\mathcal{H}$ that plays the role of bipartite graphs, with the additional condition that the annotated extension Π' is "$\mathcal{H}$-nice"'; cf. Theorem 9.4.1 for the details.

Applications. We then apply this general framework to obtain parameterized algorithms for several problems parameterized by bipartite treewidth. For each of K_t-Subgraph-Cover (Subsection 9.5.1), Weighted Vertex Cover /Independent Set (Subsection 9.5.2), Odd Cycle Transversal (Subsection 9.5.3), and Maximum Weighted Cut (Subsection 9.5.4), we prove that the problem has an annotated extension that is 1) nice and 2) solvable in FPT-time parameterized by oct, as discussed above.

To prove that an annotated problem has a nice reduction, we essentially use two ingredients. Given two compatible boundaried graphs $\mathbf{F}$ and $\mathbf{G}$ with boundary X (a boundaried graph is essentially a graph along with some labeled vertices that form a boundary, see the formal definition in Subsection 9.4.2), an annotated problem is usually nice if the following hold:

- (*Gluing property*) Given that we have guessed the annotation $\mathcal{X}$ in the boundary X, a solution compatible with the annotation is optimal in the graph $\mathbf{F} \oplus \mathbf{G}$ obtained by gluing $\mathbf{F}$ and $\mathbf{G}$ if and only if it is optimal in each of the two glued graphs. In this case, it means that the optimum on $(\mathbf{F} \oplus \mathbf{G}, \mathcal{X})$ is equal to the optimum on $(F, \mathcal{X})$ modulo some constant depending only on G and $\mathcal{X}$.
- (*Gadgetization*) Given that we have guessed the annotation in the boundary $X \setminus \{v\}$ for some vertex v in X, there is a small boundaried graph G', that is bipartite (maybe empty), such that

the optimum on $(\mathbf{F} \oplus \mathbf{G}, \mathcal{X})$ is equal to the optimum on $(\mathbf{F} \oplus \mathbf{G}', \mathcal{X})$ modulo some constant depending only on G and $\mathcal{X}$.

The gluing property seems critical to show that a problem is nice. This explains why we solve H-Subgraph-Cover only when H is a clique. For any graph H, Annotated H-Subgraph-Cover is defined similarly to Annotated Weighted Vertex Cover by specifying vertices that must or must not be taken in the solution. If H is a clique, then we crucially use the fact that H is a subgraph of $\mathbf{F} \oplus \mathbf{G}$ if and only if it is a subgraph of either F or G to prove that Annotated H-Subgraph-Cover has the gluing property. However, we observe that if H is not a clique, then Annotated H-Subgraph-Cover does not have the gluing property (see Lemma 9.5.2). This is the main difficulty that we face to solve H-Subgraph-Cover in the general case.

Note also that if we define in a similar fashion the annotated extension of Odd Cycle Transversal (that is, a set S of vertices contained in the solution and a set R of vertices that do not belong to the solution), then we can prove that this annotated extension does not have the gluing property. However, if we define Annotated Odd Cycle Transversal as the problem that takes as input a graph G and three disjoint sets S, X_1, X_2 of vertices of G and aims at finding an odd cycle transversal $S^\star$ of minimum size such that $S \subseteq S^\star$ and X_1 and X_2 are on different sides of the bipartition obtained after removing $S^\star$, then Annotated Odd Cycle Transversal does have the gluing property (see Lemma 9.5.14).

For Maximum Weighted Cut, the annotation is pretty straightforward: we use two annotation sets X_1 and X_2, corresponding to the vertices that will be on each side of the cut. It is easy to see that this annotated extension has the gluing property.

Finding the right gadgets is the main difficulty to prove that a problem is nice. As explained above, for Annotated Weighted Vertex Cover, we replace the boundaried graph $\mathbf{G}$ by an edge that simulates the behavior of G with respect to v, which is the only vertex that interest us (see Lemma 9.5.10). For Annotated K_t-Subgraph-Cover, if $\mathcal{X} = (R, S)$, depending on the optimum on $(G, (R \cup \{v\}, S))$ and the one on $(G, (R, S \cup \{v\}))$, we can show that the optimum on $(\mathbf{F} \oplus \mathbf{G}, \mathcal{X})$ is equal to the optimum on $(F, \mathcal{X})$ or $(F - v, \mathcal{X})$ modulo some constant (see Lemma 9.5.3). For Annotated Odd Cycle Transversal, if $\mathcal{X} = (S, X_1, X_2)$, we can show that the optimum on $(\mathbf{F} \oplus \mathbf{G}, \mathcal{X})$ is equal modulo some constant to the optimum on either $(F, \mathcal{X})$, or $(F - v, \mathcal{X})$, or $(F', \mathcal{X})$, where F' is obtained from F by adding an edge between v and either a vertex of X_1 or a vertex of X_2 (see Lemma 9.5.15).

Finally, let us now mention some particular ingredients used to prove that the considered annotated problems are solvable in time $f(|X|) \cdot n^{\mathcal{O}(1)}$ on an instance $(G, \mathcal{X})$ such that $G - X$ is bipartite, where $\mathcal{X}$ is a partition of a vertex set X corresponding to the annotation. For Annotated K_t-Subgraph-Cover and Annotated Weighted Vertex Cover, this is simply a reduction to (Weighted) Vertex Cover on bipartite graphs. For Odd Cycle Transversal, we adapt the algorithm of Reed, Smith, and Vetta [254] that uses iterative compression to solve Annotated Odd Cycle Transversal in FPT-time parameterized by oct, so that it takes annotations into account (Lemma 9.5.17). As for Maximum Weighted Cut parameterized by oct, the most important trick is to reduce to a K_5-odd-minor-free graph, and then use known results of Grötschel and Pulleyblank [156] and Guenin [158] to solve the problem in polynomial time (Proposition 9.5.24).

9.2 Equivalent definitions of odd-minors

In this section, we give several equivalent definitions of odd-minors.

H-expansions. Let G and H be two graphs. An *H-expansion* in G is a function η with domain $V(H) \cup E(H)$ such that:

- for every $v \in V(H)$, $\eta(v)$ is a subgraph of G that is a tree T_v, called *node* of η, such that each leaf of T_v is adjacent to a vertex of another node of η, and $\eta(v)$ is disjoint from $\eta(w)$ for distinct $v, w \in V(H)$, and
- for every $uv \in E(H)$, $\eta(uv)$ is an edge $u'v'$ in G, called *edge* of η, such that $u' \in V(\eta(u))$ and $v' \in V(\eta(v))$.

We denote by $\bigcup \eta$ the subgraph $\bigcup_{x \in V(H) \cup E(H)} \eta(x)$ of G. Given a cycle C in H, we set $\eta(C)$ to be the unique cycle in G intersecting exactly the edges $\eta(uv)$ of η such that $uv \in E(C)$.

If there is an H-expansion η in G, then notice that H is a *minor* of G given that the nodes of η form a model of H in G.

Lemma 9.2.1. *Let G and H be two graphs. The following statements are equivalent.*

1. *There is an H-expansion η in G and a 2-coloring of $\bigcup \eta$ that is proper in each node of η and such that each edge of η is monochromatic.*
2. *There is an H-expansion η in G such that every cycle in $\bigcup \eta$ has an even number of edges in $\bigcup_{v \in V(H)} \eta(v)$.*
3. *There is an H-expansion η in G such that the length of every cycle C in H has the same parity as the length of the cycle $\eta(C)$ in $\bigcup \eta$.*
4. *H can be obtained from a subgraph of G by contracting each edge of an edge cut.*

Proof. See Figure 9.1 to get some intuition.

1 $\Rightarrow$ 2: Let η be an H-expansion in G with a 2-coloring c of $\bigcup \eta$ that is proper in each node of η and such that each edge of $\bigcup \eta$ is monochromatic. $\bigcup \eta$ is a subgraph of G. The edges of $\bigcup_{v \in V(H)} \eta(v)$ are exactly the bichromatic edges of η. Let C be a cycle in $\bigcup \eta$. We transform C into a directed cyclic graph C'. The bichromatic edges in C' have alternatively color 1-2 and color 2-1. Thus, since C' is a cycle, the number of edges 1-2 and 2-1 is equal. Hence, the number of bichromatic edges in C is even.

2 $\Rightarrow$ 1: Let η be an H-expansion in G such that every cycle in $\bigcup \eta$ has an even number of edges in $\bigcup_{v \in V(H)} \eta(v)$. Let v be an arbitrary vertex of H. We color $\eta(v)$ greedily to obtain a proper 2-coloring of the node. Since $\eta(v)$ is a tree, there is only one proper 2-coloring up to isomorphism. We extend this isomorphism greedily to the entire $\bigcup \eta$ so that each edge of η is monochromatic and each node of η is properly 2-colored. Assume that there is a vertex v of $\bigcup \eta$ that is not colorable by this greedy approach. Then v is part of a cycle C in η such that each other vertex of C is colored, but the neighbors u and w of v in C give contradictory instructions for the coloring of v. If u has color c_u and w has color c_w with $c_u \neq c_w$ (resp. $c_u = c_w$), then, given that C has an even number of bichromatic edges, this implies that exactly one of uv and vw is bichromatic (resp. uv and vw are either both monochromatic or both bichromatic). Thus, v can be colored greedily. Therefore, η is an H-expansion in G with a 2-coloring c of $\bigcup \eta$ that is proper in each node of η and such that each edge of η is monochromatic.

2 $\Leftrightarrow$ 3: Let η be an H-expansion in G. By definition, there is a one-to-one correspondence between the edges of η and the edges of H. Therefore, C is a cycle of H if and only if $\eta(C)$ is a cycle of $\bigcup \eta$, and there are as many edges of η in $\eta(C)$ as the number of edges in C. The other edges of $\eta(C)$ are in the nodes of η, i.e., in $\bigcup_{v \in V(H)} \eta(v)$. Thus, every cycle in $\bigcup \eta$ has an even number of edges in $\bigcup_{v \in V(H)} \eta(v)$ if and only if the length of every cycle C in H has the same parity as the length of the cycle $\eta(C)$ in $\bigcup \eta$.

1 ⇒ 4: Let η be an H-expansion in G with a 2-coloring c of $\bigcup \eta$ that is proper in each node of η and such that each edge of η is monochromatic. $\bigcup \eta$ is a subgraph of G. Let X_1 and X_2 be the sets of vertices of η with color 1 and 2 respectively. Then $E' = E(X_1, X_2)$ is an edge cut of $\bigcup \eta$ and by contracting E' in $\bigcup \eta$, we obtain H.

4 ⇒ 1: Let G' be a subgraph of G and E' be an edge cut of G' such that H can be obtained by contracting E' in G'. Let $E'' = E(G') \setminus E'$. Let G'' be a graph obtained from G' by removing edges in E' such that every connected component of $G'' - E''$ is a spanning tree. Then there is an H-expansion η in G such that $\bigcup \eta = G''$ and the edges of η are exactly the edges in E''. Let (X_1, X_2) be the partition of $V(G')$ witnessing the edge cut E'. Then, if we give color 1 to the vertices of X_1 and color 2 to the vertices of X_2, there is a proper 2-coloring of every node of η and each edge of η is monochromatic. □

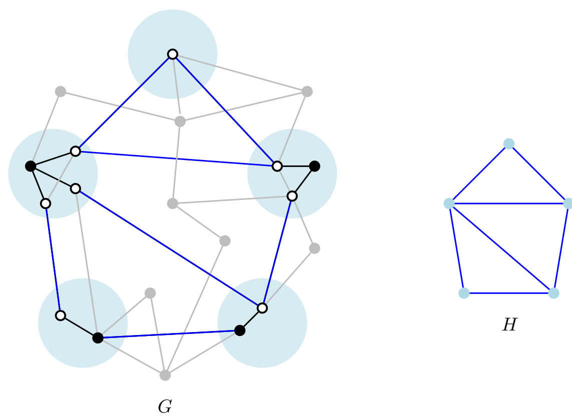

Figure 9.1: An odd H-expansion η in a graph G. The nodes of η are the subgraphs in the blue disks, and the edges of η are the blue edges in G.

An H-expansion for which one of the statements of Lemma 9.2.1 is true is called an *odd H-expansion* (see Figure 9.1 for an illustration). Using Statement 3, we say that such an H-expansion in G *preserves cycle parity.* If there is an H-expansion in G, then we say that H is an *odd-minor* of G. Note that if H is an odd-minor of G, then H is a minor of G. However, the opposite does not always hold. For instance, K_3 is a minor of C_4, but is not an odd-minor of C_4. We say that a graph G is *H-odd-minor-free* if G excludes the graph H as an odd-minor. In particular, observe that bipartite graphs are exactly the K_3-odd-minor-free graphs and that the forests are exactly the $\{K_3, C_4\}$-odd-minor-free graphs.

9.3 Bipartite treewidth

Let us define formally bipartite treewidth and the more general parameter 1-$\mathcal{H}$-treewidth, and give some preliminary results.

Notations on tree decompositions. Let (T, χ, r) be a rooted tree decomposition. Given $t \in V(T)$, we denote by $\mathsf{Ch}_r(t)$ the set of children of t and by $\mathsf{Par}_r(t)$ the parent of t (if $t \neq r$). We set $\delta_t^r = \mathsf{adh}(t, \mathsf{Par}_r(t))$, with the convention that $\delta_r^r = \emptyset$. Moreover, we denote by G_t^r the graph induced by $\bigcup_{t' \in V(T_t)} \chi(t')$ where (T_t, t) is the rooted subtree of (T, r) containing all descendants of t. We may use δ_t and G_t instead of δ_t^r and G_t^r when there is no risk of confusion.

While our goal in this chapter is to study bipartite treewidth, defined below, we provide the following definition in a more general way, namely of a parameter that we call 1-$\mathcal{H}$-treewidth, with the hope of it finding some application in future work. We use the term 1-$\mathcal{H}$-treewidth to signify that the $\mathcal{H}$-part of each bag intersects each neighboring bag in at most one vertex. This also has the benefit of avoiding confusion with $\mathcal{H}$-treewidth defined in [104], which would be another natural name for this class of parameters.

1-$\mathcal{H}$-treewidth. Let $\mathcal{H}$ be a graph class. A *1-$\mathcal{H}$-tree decomposition* of a graph G is a triple (T, α, β), where T is a tree and $\alpha, \beta : V(T) \to 2^{V(G)}$, such that

- $(T, \alpha \cup \beta)$ is a tree decomposition of G, where $\alpha \cup \beta$ maps each $t \in V(T)$ to $\alpha(t) \cup \beta(t)$,
- for every $t \in V(T)$, $\alpha(t) \cap \beta(t) = \emptyset$,
- for every $t \in V(T)$, $G[\beta(t)] \in \mathcal{H}$, and
- for every $tt' \in E(T)$, $|(\alpha \cup \beta)(t') \cap \beta(t)| \leq 1$.

The vertices in $\alpha(t)$ are called *apex vertices* of the node $t \in V(T)$.

The *width* of (T, α, β) is equal to $\max\{\, |\alpha(t)| \;\big|\; t \in V(T)\}$. The *1-$\mathcal{H}$-treewidth* of G, denoted by $(1, \mathcal{H})\text{-}\mathsf{tw}(G)$, is the minimum width over all 1-$\mathcal{H}$-tree decompositions of G.

A *rooted 1-$\mathcal{H}$-tree decomposition* is a tuple (T, α, β, r) where (T, α, β) is a 1-$\mathcal{H}$-tree decomposition and (T, r) is a rooted tree.

Given that $(T, \alpha \cup \beta)$ is a tree decomposition, we naturally extend all definitions and notations concerning treewidth to 1-$\mathcal{H}$-treewidth.

Observe also that a tree decomposition (T, χ) is also a 1-$\mathcal{H}$-tree decomposition for every graph class $\mathcal{H}$, in the sense that (T, χ, o) is a 1-$\mathcal{H}$-tree decomposition, where $o : V(T) \to \emptyset$. Therefore, for every graph class $\mathcal{H}$ and every graph G, $(1, \mathcal{H})\text{-}\mathsf{tw}(G) \leq \mathsf{tw}(G) + 1$.

If $\mathcal{H}$ is the graph class containing only the empty graph, then a 1-$\mathcal{H}$-tree decomposition is exactly a tree decomposition.

Remark. The $\mathcal{H}$-treewidth actually corresponds to the 0-$\mathcal{H}$-treewidth (minus one), which is defined by replacing the "1" by a "0" in the last item of the definition of a 1-$\mathcal{H}$-tree decomposition above. Indeed, let (T, α, β) be a 0-$\mathcal{H}$-tree decomposition of a graph G of width k. Note that, for each distinct $t, t' \in V(T)$, $\beta(t) \cap \beta(t') = \emptyset$. Let $X = \bigcup_{t \in V(T)} \alpha(t)$. Then (T, α) is a tree decomposition of X of width $k - 1$. Moreover, for each $t \in V(T)$, $G[\beta(t)] \in \mathcal{H}$, and therefore the connected components of $G - X$ belong to $\mathcal{H}$.

Bipartite treewidth [adapted from [85, 299]]. A graph G is *bipartite* if there is a partition (A, B) of $V(G)$ such that $E(G) = E(A, B)$. We denote the class of bipartite graphs by $\mathcal{B}$. We focus here on the case where $\mathcal{H} = \mathcal{B}$. Then, we use the term *bipartite treewidth* instead of 1-$\mathcal{H}$-treewidth, and denote it by btw. As mentioned in the introduction, this definition had already been used (more or less implicitly) in [85, 299].

Given that the bipartite graphs are closed under 1-clique-sums, we have the following.

Observation 9.3.1. *A graph has bipartite treewidth zero if and only if it is bipartite.*

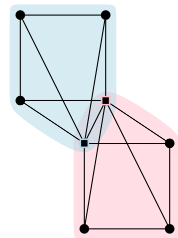

Figure 9.2: A graph of bipartite treewidth one. A corresponding bipartite tree decomposition of width one is depicted, with two bags (one blue and one pink). The apex vertex of each bag is the squared vertex of the same color.

Moreover, Campbell, Gollin, Hendrey, and Wiederrecht [49] recently announced an FPT-approximation algorithm to construct a bipartite tree decomposition.

Proposition 9.3.2 ([49]). *There exist computable functions $f_1, f_2, g : \mathbb{N} \to \mathbb{N}$ and an algorithm that, given a graph G and $k \in \mathbb{N}$, outputs, in time $g(k) \cdot n^4 \log n$, either a report that $\mathsf{btw}(G) \geq f_1(k)$, or a bipartite tree decomposition of G of width at most $f_2(k)$.*

Bipartite treewidth is not closed under minors, given that contracting an edge in a bipartite graph (which has bipartite treewidth zero) may create a non-bipartite graph (which has positive bipartite treewidth). However, bipartite treewidth is closed under odd-minors, which is a desirable property to deal with odd-minor related problems.

Lemma 9.3.3. *Bipartite treewidth is closed under odd-minor containment.*

Proof. Let G be a graph and H be an odd-minor of G. We want to prove that $\mathsf{btw}(H) \leq \mathsf{btw}(G)$. By Lemma 9.2.1, there is a subgraph G' of G and an edge cut E' such that H is obtained from G' by contracting every edge in E'.

Since we only removed vertices and edges to obtain G' from G, $\mathsf{btw}(G') \leq \mathsf{btw}(G)$. It remains to show that $\mathsf{btw}(H) \leq \mathsf{btw}(G')$. Let (T, α', β') be a bipartite tree decomposition of G'. We transform (T, α', β') to a bipartite tree decomposition (T, α, β) of H as follows. For each $e = uv \in E'$ and for each $t \in V(T)$ such that $\{u, v\} \cap (\alpha' \cup \beta')(t) \neq \emptyset$,

- if $\{u, v\} \cap \alpha'(t) \neq \emptyset$, then the vertex v_e resulting from contracting e is placed in $\alpha(t)$,
- otherwise, v_e is placed in $\beta(t)$.

For each $v \in V(G')$ that is not involved in any contraction and for each $t \in V(T)$, if $v \in \alpha'(t)$ (resp. $v \in \beta'(t)$), then $v \in \alpha(t)$ (resp. $v \in \beta(t)$).

Let us show that (T, α, β) is indeed a bipartite tree decomposition of H. It is easy to see that $(T, \alpha \cup \beta)$ is a tree decomposition of H, since it is obtained from $(T, \alpha' \cup \beta')$ by contracting the edge set E', and that treewidth is minor-closed. For simplicity, we identify the vertices in $\alpha \cup \beta$ with the vertices of H. Given that an edge with at least one endpoint in $\alpha'(t)$ contracts to a vertex in $\alpha(t)$, no new vertex is added to $\beta(t)$, and therefore, for any $t' \in V(T) \setminus \{t\}$, $|(\alpha \cup \beta)(t') \cap \beta(t)| \leq 1$.

It remains to prove that, for each $t \in V(T)$, $H[\beta(t)]$ is bipartite. Let $t \in V(T)$. Let E_t be the set of edges of E' with both endpoints in $\beta'(t)$. We have to prove that the bipartite graph induced by $\beta'(t)$ in G' remains bipartite after contracting E_t. E_t is an edge cut of $G'[\beta'(t)]$, witnessed by some vertex partition (A_1, A_2). Given a proper 2-coloring (B_1, B_2) of $G'[\beta'(t)]$, which is bipartite, keep the same color for the vertices in A_1, and change the color of the vertices in A_2, i.e., define the coloring $(C_1, C_2) = ((B_1 \cap A_1) \cup (B_2 \cap A_2), (B_2 \cap A_1) \cup (B_1 \cap A_2))$. Thus, the monochromatic edges are exactly the edges of E_t. Therefore, contracting E_t gives a proper 2-coloring of $H[\beta(t)]$, so $H[\beta(t)]$ is bipartite. Thus, (T, α, β) is a bipartite tree decomposition of H.

Moreover, since the contraction of an edge with both endpoints in $\beta'(t)$ is a vertex in $\beta(t)$, it follows that $|\alpha(t)| \leq |\alpha'(t)|$ for every $t \in V(T)$. Therefore, $\mathsf{btw}(H) \leq \mathsf{btw}(G')$. $\square$

A natural generalization of bipartite treewidth can be made by replacing the "1" in the last item of the definition of 1-$\mathcal{H}$-tree decomposition by any $q \in \mathbb{N}$, hence defining *q-$\mathcal{H}$-tree decompositions* and *q-$\mathcal{H}$-treewidth*, denoted by $(q, \mathcal{H})\text{-}\mathsf{tw}(G)$. For $q \geq 2$, however, q-$\mathcal{B}$-treewidth is not closed under odd-minor containment, as we prove in Lemma 9.3.4 below. Additionally, given that for $q \in \{0, 1\}$, $\mathsf{torso}(G - \alpha(t), \beta(t)) = G[\beta(t)]$, we could replace the third item by the property "for every $t \in V(T)$, $\mathsf{torso}(G - \alpha(t), \beta(t)) \in \mathcal{H}$", hence defining what we call *q-torso-$\mathcal{H}$-tree decompositions* and *q-torso-$\mathcal{H}$-treewidth*, denoted by $(q, \mathcal{H})^\star\text{-}\mathsf{tw}(G)$. However, we prove in Lemma 9.3.4 that, for $q \geq 2$, $(q, \mathcal{B})^\star\text{-}\mathsf{tw}$ is also not closed under odd-minors. These facts, in our opinion, provide an additional justification for the choice of $q = 1$ in the definition of bipartite treewidth.

Lemma 9.3.4. *For $q \geq 2$, q(-torso)-$\mathcal{B}$-treewidth is not closed under odd-minor containment. In particular, for any $t \geq 3$, there exist a graph G and an odd-minor H of G, such that $(q, \mathcal{B})\text{-}\mathsf{btw}(G) = 0$ and $(q, \mathcal{B})\text{-}\mathsf{btw}(H) = t - 2$, and $(q, \mathcal{B})^\star\text{-}\mathsf{btw}(G) \leq 1$ and $(q, \mathcal{B})^\star\text{-}\mathsf{btw}(H) = t - 2$.*

Proof. Let $t \in \mathbb{N}_{\geq 3}$ and let K'_t (resp. K''_t) be the graph obtained from K_t by subdividing every edge once (resp. twice). Let $V' = \{v_1, \ldots, v_t\}$ be the set of vertices of K''_t that are the original vertices of K_t. K_t is an odd-minor of K''_t since K_t can be obtained from K''_t by contracting the edge cut $E(V', V(K''_t) \setminus V')$. Note also that K'_t is bipartite. We show that taking $G = K''_t$ and $H = K_t$ satisfies the statement of the lemma.

Given that K_t is a complete graph, it has to be fully contained in one bag of any tree decomposition, so in particular of any q(-torso)-$\mathcal{B}$-tree decomposition. Since the smallest odd cycle transversal of K_t has size $t - 2$, we have that $(q, \mathcal{B})^\star\text{-}\mathsf{tw}(K_t) = (q, \mathcal{B})\text{-}\mathsf{tw}(K_t) = t - 2$.

Let us first prove that $(q, \mathcal{B})\text{-}\mathsf{tw}(K''_t) = 0$. For $i, j \in [t]$ with $i < j$, let $e_{i,j}$ be the path of length three between v_i and v_j. Let T be a tree with one central vertex x_0 and, for each $i, j \in [t]$ with $i < j$, a vertex $x_{i,j}$ only adjacent to x_0 (thus, T is a star). Let $\beta(x_0) = V'$ and $\beta(x_{i,j}) = V(e_{i,j})$ for each $i, j \in [t]$ with $i < j$. Let $\alpha(x) = \emptyset$ for each $x \in V(T)$. V' is an independent set, so $G[V']$ is bipartite. Note that paths are bipartite. Moreover, each adhesion contains at most two vertices of $\beta(x_0)$ and two vertices of $\beta(x_{i,j})$. Hence, (T, α, β) is a q-$\mathcal{B}$-tree decomposition of K''_t, for $q \geq 2$, and has width zero.

Let us now prove that $(q, \mathcal{B})^\star\text{-}\mathsf{btw}(K''_t) \leq 1$. Let $u_{i,j}$ and $w_{i,j}$ be the internal vertices of $e_{i,j}$, such that $u_{i,j}$ is adjacent to v_i. Let V_1 (resp. V_2) be the set of vertices $u_{i,j}$ (resp. $w_{i,j}$). We construct a q-torso-$\mathcal{B}$-tree decomposition (T, α', β') of K''_t as follows. We set $\alpha'(x_0) = \emptyset$ and $\beta'(x_0) = V' \cup V_2$. For each $i, j \in [t]$ with $i < j$, we set $\alpha'(x_{i,j}) = \{v_i\}$ and $\beta'(x_{i,j}) = \{u_{i,j}, w_{i,j}\}$. Observe that $\mathsf{torso}(K''_t - \alpha'(x_0), \beta'(x_0)) = K'_t$, since each path v_i-$u_{i,j}$-$w_{i,j}$ is replaced by an edge $v_i w_{i,j}$. Thus, it is bipartite. Similarly, the torso at each other node of T is an edge, and hence is bipartite as well. Moreover, each adhesion contains at most two vertices of $\beta'(x_0)$ and one vertex of $\beta'(x_{i,j})$. Hence, (T, α', β') is indeed a q-torso-$\mathcal{B}$-tree decomposition of K''_t, for $q \geq 2$, and has width one. Therefore, $(q, \mathcal{B})^\star\text{-}\mathsf{btw}(K''_t) \leq 1$.

Hence, q(-torso)-$\mathcal{B}$-treewidth is not closed under odd-minor containment and the gap between a graph and an odd-minor of this graph can be arbitrarily large. $\square$

As mentioned in Section 9.1, one of the main difficulties for doing dynamic programming on (rooted) bipartite tree decompositions is the lack of a way to upper-bound the number of children of each node of the decomposition. As shown in the next lemma, the notion of "nice tree decomposition" is not generalizable to bipartite tree decompositions.

Lemma 9.3.5. *For any $t \in \mathbb{N}$, there exists a graph G such that $\mathsf{btw}(G) = 1$ and any bipartite tree decomposition of G whose nodes all have at most t neighbors has width at least $t - 1$.*

Proof. Let G be the graph obtained from $K_{t,t}$ by gluing a new pendant triangle H_v to each vertex v of $K_{t,t}$ (that is, v is identified with one vertex of its pendant triangle). Let T be the star $K_{1,2t}$, with vertex set $\{t_0\} \cup \{t_v \mid v \in V(K_{t,t})\}$. Let $\alpha(t_0) = \emptyset$, $\beta(t_0) = V(K_{t,t})$, and for every $v \in V(K_{t,t})$, $\alpha(t_v) = \{v\}$ and $\beta(t_v) = V(H_v) \setminus \{v\}$. It can be easily verified that $\mathcal{T} = (T, \alpha, \beta)$ is a bipartite tree decomposition of G of width one. Given that G is not bipartite, Observation 9.3.1 implies that $\mathsf{btw}(G) = 1$. Note that node t_0 has $2t$ neighbors. For any bipartition (A, B) of $V(K_{t,t})$ such that $A, B \neq \emptyset$, we have $|E(A, B)| \geq t$. Hence, for any bipartite tree decomposition $\mathcal{T}'$ of G such that $V(K_{t,t})$ is not totally contained in one bag, there is an adhesion of two bags of size at least t, so the width of $\mathcal{T}'$ is at least $t - 1$. If $V(K_{t,t})$ is fully contained in one bag, however, the only way to reduce the number of children to k, for some integer k, is to add $2t - k$ of the pendant triangles inside the same bag. But then this bag has odd cycle transversal number at least $2t - k$, so the obtained bipartite tree decomposition has width at least $2t - k$. Hence, if we want that $k \leq t$, then the corresponding width is at least $2t - k \geq t$. $\square$

9.4 General dynamic programming to obtain FPT-algorithms

In this section, we introduce a framework for obtaining FPT-algorithms for problems parameterized by the width of a given bipartite tree decomposition of the input graph. In Subsection 9.4.1 we define several gluing operations on boundaried graphs that are needed for our scheme. In Subsection 9.4.2 we introduce the main technical notion of a *nice problem* and the necessary background, and in Subsection 9.4.3 we provide dynamic programming algorithms for nice problems.

9.4.1 Gluing boundaried graphs

Let us define ways to glue boundaried graphs that will be used in this chapter.

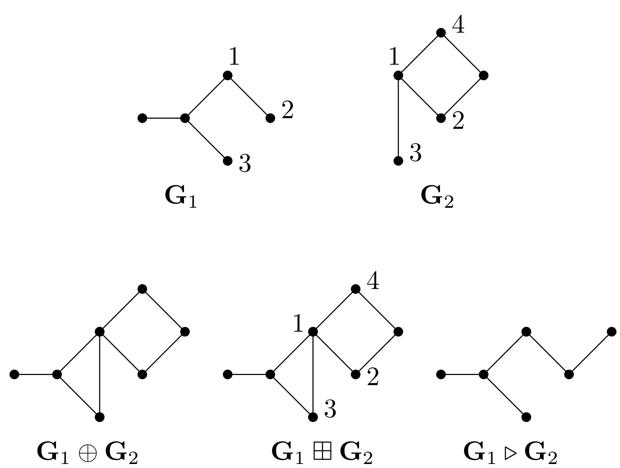

Figure 9.3: Examples of use of the operations $\oplus$, $\boxplus$, and $\triangleright$.

Just for this chapter, we redefine the $\oplus$ operation as follows. Given two boundaried graphs $\mathbf{G}_1 = (G_1, B_1, \rho_1)$ and $\mathbf{G}_2 = (G_2, B_2, \rho_2)$, we define $\mathbf{G}_1 \oplus \mathbf{G}_2$ as the unboundaried graph obtained if we take the disjoint union of G_1 and G_2 and, for every $i \in \rho_1(B_1) \cap \rho_2(B_2)$, we identify vertices $\rho_1^{-1}(i)$ and $\rho_2^{-1}(i)$. Note that we do not ask $\mathbf{G}_1$ and $\mathbf{G}_2$ to be compatible (for this chapter). If two vertices are adjacent in at least one boundary, then there are adjacent in $\mathbf{G}_1 \oplus \mathbf{G}_2$. If v is the result of the identification of $v_1 := \rho_1^{-1}(i)$ and $v_2 := \rho_2^{-1}(i)$ then we say that v is the *heir* of v_i from $\mathbf{G}_i, i \in [2]$. If v is either a vertex in B_1 where $\rho_1(v) \notin \rho_1(B_1) \cap \rho_2(B_2)$ or a vertex in B_2 where $\rho_2(v) \notin \rho_1(B_1) \cap \rho_2(B_2)$, then v is also a (non-identified) vertex of $\mathbf{G}_1 \oplus \mathbf{G}_2$ and is an *heir* of itself (from $\mathbf{G}_1$ or $\mathbf{G}_2$ respectively). For $i \in [2]$, and given an edge vu in $\mathbf{G}_1 \oplus \mathbf{G}_2$, we say that vu is the *heir* of an edge $v'u'$ from $\mathbf{G}_i$ if v' is the heir of v from $\mathbf{G_i}$, u' is the heir of u from $\mathbf{G_i}$, and $v'u'$ is an edge of G_i. If x' is an heir of x from $\mathbf{G} = (G, B, \rho)$ in $\mathbf{G}'$, then we write $x = \mathsf{heir}_{\mathbf{G},\mathbf{G}'}(x')$. Given $B' \subseteq B$, we write $\mathsf{heir}_{\mathbf{G},\mathbf{G}'}(B') = \bigcup_{v \in B'} \mathsf{heir}_{\mathbf{G},\mathbf{G}'}(x')$.

We also define $\mathbf{G}_1 \boxplus \mathbf{G}_2$ as the *boundaried* graph $(\mathbf{G}_1 \oplus \mathbf{G}_2, B, \rho)$, where B is the set of all heirs from $\mathbf{G_1}$ and $\mathbf{G_2}$ and $\rho : B \to \mathbb{N}$ is the union of ρ_1 and ρ_2 after identification. Note that in circumstances where $\boxplus$ is repetitively applied, the heir relation is maintained due to its transitivity.

Moreover, we define $\mathbf{G}_1 \triangleright \mathbf{G}_2$ as the unboundaried graph G obtained from $\mathbf{G}_1 \oplus \mathbf{G}_2$ by removing all heirs from $\mathbf{G}_2$ that are not heirs from $\mathbf{G}_1$ and all heirs of edges from $\mathbf{G}_2$ that are not heirs of edges from $\mathbf{G}_1$. Informally, $\mathbf{G}_1 \triangleright \mathbf{G}_2$ is obtained from $\mathbf{G}_1 \oplus \mathbf{G}_2$ by removing vertices and edges of $G[B_2]$ that are not in $G[B_1]$. Note that $\triangleright$ is not commutative. See Figure 9.3 for an illustration of the operations $\oplus$, $\boxplus$, and $\triangleright$.

9.4.2 Nice problems

All algorithms we give for problems on graphs of bounded btw follow the same strategy. To avoid unnecessary repetition, we introduce a framework that captures the features that the problems have in common with respect to their algorithms using bipartite tree decompositions. Naturally, the algorithms use dynamic programming along a rooted bipartite tree decomposition. However, as the bags in the decomposition can now be large, we cannot apply brute-force approaches to define table entries as we can, for instance, on standard tree decompositions when dealing with treewidth.

Suppose we are at a node t with children $t_1, \ldots, t_d$. Since the size of $\alpha(t)$ is bounded by the width, we can store all possible ways in which solutions interact with $\alpha(t)$. Moreover, since each adhesion has at most one vertex from $\beta(t)$, the size of each adhesion is still bounded in terms of the bipartite treewidth. Therefore, we can store one table entry for each way in which a solution can interact with the adhesion δ_t of t and its parent. However, since the size of $\beta(t)$ is unbounded, there can now be an exponential (in n) number of choices of table entries that are "compatible" with the choice $\mathcal{X}$ made for δ_t, so we cannot simply brute-force them to determine the optimum value corresponding to $\mathcal{X}$. To overcome this, we apply the following strategy: First, since the size of $\alpha(t)$ is bounded in terms of the bipartite treewidth, we guess which choice $\mathcal{A}$ of the interaction of the solution with $\alpha(t) \cup \delta_t$ that extends $\mathcal{X}$ leads to the optimum partial solution. For each $i \in [d]$, there may be a vertex $v_{t_i} \in \delta_{t_i} \cap \beta(t)$ whose interaction with the partial solution remained undecided. We replace, for each $i \in [d]$, the subgraph $G_{t_i} - \delta_{t_i}$ with a simply structured subgraph that simulates the behaviour of the table at t_i when it comes to the decision of how v_{t_i} interacts with the solution, under the choice of $\mathcal{A}$ for $\alpha(t) \cup \delta_t$. The crux is that the resulting graph will have an odd cycle transversal that is bounded in terms of the size of $\alpha(t)$, so we can apply known FPT-algorithms parameterized by odd cycle transversal to determine the value of the table entry. These notions can be formalized not only for bipartite treewidth, but for any 1-$\mathcal{H}$-treewidth, so we present them in full generality here. We also depart from using tree decompositions explicitly, and state them in an equivalent manner in the language of boundaried graphs.

First, let us formalize the family of problems we consider which we refer to as *optimization problems*. Here, solutions correspond in some sense to partitions of the vertex set, and we want to optimize some property of such a partition. For instance, if we consider Odd Cycle Transversal, then this partition has three parts, one for the vertices in the solution, and one part for each part of the bipartition of the vertex set of the graph obtained by removing the solution vertices, and we want to minimize the size of the first part of the partition. (It will become clear later why we keep one separate part for each part of the bipartition.) In Maximum Cut, the partition simply points to which side of the cut each vertex is on, and we want to maximize the number of edges going across.

A *p-partition-evaluation function on graphs* is a function f that receives as input a graph G along with a p-partition $\mathcal{P}$ of its vertices and outputs a non-negative integer. Given such a function f and some choice $\mathsf{opt} \in \{\max, \min\}$ we define the associated graph parameter $\mathsf{p}_{f,\mathsf{opt}}$ where, for every graph G,

$$\mathsf{p}_{f,\mathsf{opt}}(G) = \mathsf{opt}\{f(G, \mathcal{P}) \mid \mathcal{P} \text{ is a } p\text{-partition of } V(G)\}.$$

An *optimization problem* is a problem that can be expressed as follows.

Input:	A graph G.
Task:	Compute $\mathsf{p}_{f,\mathsf{opt}}(G)$.

To represent the case when we made some choices for the (partial) solution to an optimization problem, such as $\mathcal{A}$ above, we consider *annotated* versions of such problems. They extend the function $\mathsf{p}_{f,\mathsf{opt}}$ so to receive as input, apart from a graph, a set of annotated sets in the form of a partition $\mathcal{X} \in \mathcal{P}_p(X)$ of some $X \subseteq V(G)$. More formally, the *annotated extension* of $\mathsf{p}_{f,\mathsf{opt}}$ is the parameter $\hat{\mathsf{p}}_{f,\mathsf{opt}}$ such that

$$\hat{\mathsf{p}}_{f,\mathsf{opt}}(G, \mathcal{X}) = \mathsf{opt}\{f(G, \mathcal{P}) \mid \mathcal{P} \text{ is a } p\text{-partition of } V(G) \text{ with } \mathcal{X} \subseteq \mathcal{P}\}.$$

Observe that $\mathsf{p}_{f,\mathsf{opt}}(G) = \hat{\mathsf{p}}_{f,\mathsf{opt}}(G, \emptyset^p)$, for every graph G. The problem Π' is a *p-annotated extension* of the optimization problem Π if Π can be expressed by some p-partition-evaluation function f and some choice $\mathsf{opt} \in \{\max, \min\}$, and if Π' can be expressed as follows.

Input:	A graph G and $\mathcal{X} \in \mathcal{P}_p(X)$ for some $X \subseteq V(G)$.
Task:	Compute $\hat{\mathsf{p}}_{f,\mathsf{opt}}(G, \mathcal{X})$.

We also say that Π' is a *p-annotated problem.*

Let us turn to the main technical tool introduced in this section that formalizes the above idea, namely the *nice reduction*. First, we may assume that the vertices of G are labeled injectively via σ. Then, each graph G_{t_i}, for $i \in [d]$, naturally corresponds to a boundaried graph $\mathbf{G_i} = (G_{t_i}, \delta_{t_i}, \sigma|_{\delta_{t_i}})$; from now on let $X_i = \delta_{t_i}$. The part of G_t that will be modified can be viewed as a boundaried graph $\mathbf{G}$ which is essentially obtained as $\boxplus_{i\in[d]}\mathbf{G_i}$. However, as we want to fix a choice of how the partial solution interacts with δ_t, we include these corresponding vertices in $\mathbf{G}$ as well, modeled as a trivial boundaried graph $\mathbf{X}$, making $\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]}\mathbf{G_i})$.

Denote the boundary of $\mathbf{G}$ by X. The set X is partitioned into (A, B), corresponding to $(X \cap (\alpha(t) \cup \delta(t)), X \cap (\beta(t) \setminus \delta(t)))$, and the fact that the adhesion between t_i and t had at most one vertex in common with $\beta(t)$ now materializes as the fact that for each $i \in [d]$, $\mathbf{G}$ has at most one vertex outside of A that is an heir of a vertex in $\mathbf{G_i}$. Fixing a choice for $\alpha(t) \cup \delta_t$ now corresponds to choosing a partition $\mathcal{A}$ of the set A. As we assume that all table entries at the children have been computed, we assume knowledge of all values $\hat{\mathsf{p}}_{f,\mathsf{opt}}(G_{t_i}, \mathcal{X}_i)$, for all $i \in [d]$ and $\mathcal{X}_i \in \mathcal{P}_p(X_i)$. This finishes the motivation of the input of a nice reduction.

Given the pair $(\mathbf{G}, \mathcal{A})$, it outputs a tuple $(\mathbf{G'} = (G', X', \rho'), \mathcal{A}', s')$, with the following desired properties. $\mathbf{G'}$ can be constructed by gluing d' boundaried graphs plus one trivial one (for some

$d' \in \mathbb{N}$), similarly to $\mathbf{G}$. $\mathcal{A}'$ is a p-partition of a set $A' \subseteq V(G')$ whose size is at most the size of A plus a constant. No matter what the structure of the graph of the vertices in $(\alpha \cup \beta)(t)$ looked like (remember, so far we carved out only the adhesions), the solutions are preserved, up to an offset of s'. This is modeled by saying that for each boundaried graph $\mathbf{F}$ (which corresponds to the remainder of the bag at t) compatible with $\mathbf{G}$, $\hat{\mathsf{p}}_{f,\mathsf{opt}}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f,\mathsf{opt}}(\mathbf{G}' \triangleright \mathbf{F}, \mathcal{A}') + s'$. The reason why we use the $\triangleright$-operator in the right-hand side of the equation is the gadgeteering happening in the later sections. To achieve the "solution-preservation", we might have to add or remove vertices, or change adjacencies between vertices in X_i.

The last condition corresponds to our aim that if the bag at t induces a graph of small oct (now, a small modulator to a graph class $\mathcal{H}$), then the entire graph resulting from the operation $(\mathbf{G}' \triangleright \mathbf{F})$ should have a small modulator to $\mathcal{H}$ (namely A'). All remaining conditions are related to the efficiency of the nice reduction.

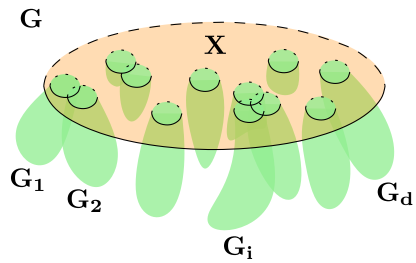

Figure 9.4: Illustration of $\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i)$.

Nice problem and nice reduction. Let $p \in \mathbb{N}$, let $\mathcal{H}$ be a graph class, and let Π be a p-annotated problem corresponding to some choice of a p-partition-evaluation function f and some $\mathsf{opt} \in \{\max, \min\}$. We say that Π is a *$\mathcal{H}$-nice problem* if there exists an algorithm that receives as input

- a boundaried graph $\mathbf{G} = (G, X, \rho)$,
- a trivial boundaried graph $\mathbf{X} = (G[X], X, \rho_X)$ and a collection $\{\mathbf{G}_i \mid i \in [d]\}$ of boundaried graphs with $\mathbf{G}_i = (G_i, X_i, \rho_i)$ for $i \in [d]$, such that $d \in \mathbb{N}$ and $\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i)$ (see Figure 9.4),
- a partition (A, B) of X such that for all $i \in [d]$, $|\mathsf{heir}_{\mathbf{G_i},\mathbf{G}}(X_i) \setminus A| \leq 1$,
- some $\mathcal{A} \in \mathcal{P}_p(A)$, and
- for every $i \in [d]$ and each $\mathcal{X}_i \in \mathcal{P}_p(X_i)$, the value $\hat{\mathsf{p}}_{f,\mathsf{opt}}(G_i, \mathcal{X}_i)$,

and outputs, in time $\mathcal{O}(|A| \cdot d)$, a tuple $(\mathbf{G}' = (G', X', \rho'), \mathcal{A}', s')$, called *$\mathcal{H}$-nice reduction of the pair* $(\mathbf{G}, \mathcal{A})$ *with respect to* Π, such that the following hold (see Figure 9.5 for an illustration).

- There are sets $A' \subseteq V(G')$ and $\mathcal{A}' \in \mathcal{P}_p(A')$ with $|A'| = |A| + \mathcal{O}(1)$.
- There is a trivial boundaried graph $\mathbf{X}' = (G[X'], X', \rho_{X'})$ and a collection $\{\mathbf{G}'_i = (G'_i, X'_i, \rho'_i) \mid i \in [d']\}$, where $d' \in \mathbb{N}$, of boundaried graphs such that $\mathbf{G}' = \mathbf{X}' \boxplus (\boxplus_{i\in[d']} \mathbf{G}'_i)$ and $|V(G')| \leq |X| + \mathcal{O}(|B|)$, $|E(G')| \leq |E(G[X])| + \mathcal{O}(|B|)$.
- For any boundaried graph $\mathbf{F}$ compatible with $\mathbf{G}$, it holds that

$$\hat{\mathsf{p}}_{f,\mathsf{opt}}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) \quad = \quad \hat{\mathsf{p}}_{f,\mathsf{opt}}(\mathbf{G}' \triangleright \mathbf{F}, \mathcal{A}') + s'.$$

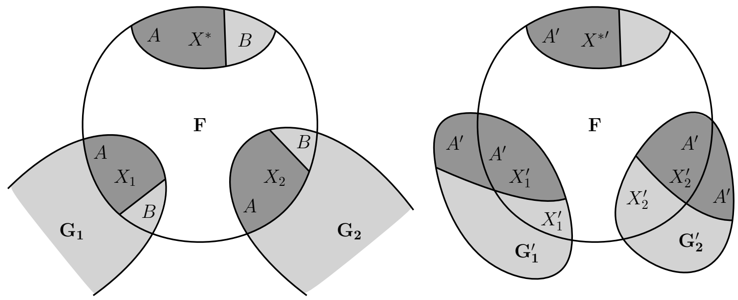

Figure 9.5: Illustration of the setting of the nice problem and reduction. The shaded area on the left is $\mathbf{G}$ where $X = X_1 \cup X_2 \cup X^\star$, and the shaded area on the right is $\mathbf{G}'$ where $X' = X_1' \cup X_2' \cup X^{\star\prime}$.

- For any boundaried graph $\mathbf{F} = (F, X_F, \rho_F)$ compatible with $\mathbf{G}$, if $\bar{F} - A_F \in \mathcal{H}$, where $\bar{F} = (\mathbf{F} \oplus \mathbf{G})[\mathsf{heir}_{\mathbf{F},\mathbf{G}\oplus\mathbf{F}}(V(F))]$ and $A_F = \mathsf{heir}_{\mathbf{G},\mathbf{G}\oplus\mathbf{F}}(A)$, then $(\mathbf{G}' \triangleright \mathbf{F}) - A' \in \mathcal{H}$.

All the definitions of this section are naturally generalizable to graphs with weights on the vertices and/or edges. Given such a weight function w, we extend $f(G, \mathcal{P})$, $\mathsf{p}_{f\mathsf{opt}}(G)$, $\hat{\mathsf{p}}_{f\mathsf{opt}}(G, \mathcal{X})$, $(\mathbf{G}, \mathcal{A})$, and $(\mathbf{G}', \mathcal{A}', s')$ to $f(G, \mathcal{P}, w)$, $\mathsf{p}_{f\mathsf{opt}}(G, w)$, $\hat{\mathsf{p}}_{f\mathsf{opt}}(G, \mathcal{X}, w)$, $(\mathbf{G}, \mathcal{A}, w)$, and $(\mathbf{G}', \mathcal{A}', s', w')$, respectively.

9.4.3 General dynamic programming scheme

We now have all the ingredients for our general dynamic programming scheme on bipartite tree decompositions. We essentially prove that if a problem Π has an annotated extension that is $\mathcal{B}$-nice and solvable in FPT-time parameterized by oct, then Π is solvable in FPT-time parameterized by btw. This actually holds for more general $\mathcal{H}$.

Theorem 9.4.1. *Let $p \in \mathbb{N}$. Let $\mathcal{H}$ be a graph class. Let Π be an optimization problem. Let Π' be a problem that is:*

- *a p-annotated extension of Π corresponding to some choice of p-partition-evaluation function g and some $\mathsf{opt} \in \{\max, \min\}$,*
- *$\mathcal{H}$-nice, and*
- *solvable, given instances $(G, \mathcal{X})$ such that $G - \cup\mathcal{X} \in \mathcal{H}$, in time $f(|\cup\mathcal{X}|) \cdot n^c \cdot m^d$, for some $c, d \in \mathbb{N}$.*

Then, there is an algorithm that, given a graph G and a 1-$\mathcal{H}$-tree decomposition of G of width k, computes $\mathsf{p}_{f,\mathsf{opt}}(G)$ in time $\mathcal{O}(p^k \cdot f(k + \mathcal{O}(1)) \cdot (k \cdot n)^c \cdot m^d)$ (or $\mathcal{O}(p^k \cdot f(k + \mathcal{O}(1)) \cdot (m + k^2 \cdot n)^d)$ if $c = 0$).

Proof. Let Alg be the algorithm that solves instances $(G, \mathcal{X})$ such that $G - \cup\mathcal{X} \in \mathcal{H}$ in time $f(|\cup\mathcal{X}|) \cdot n^c \cdot m^d$.

Let (T, α, β, r) be a rooted 1-$\mathcal{H}$-tree decomposition of G of width at most k. Let $\sigma : V(G) \to \mathbb{N}$ be an injective function. For $t \in V(T)$, we set the following:

$$\mathbf{G}_t = (G_t, \delta_t, \sigma|_{\delta_t}), \qquad X_t = \alpha(t) \cup \delta_t \cup \bigcup_{t' \in \mathsf{Ch}_r(t)} \delta_{t'},$$

$$\mathbf{X}_t = (G[X_t], X_t, \sigma|_{X_t}), \qquad \mathbf{H}_t = \mathbf{X}_t \boxplus (\boxplus_{t' \in \mathsf{Ch}_r(t)} \mathbf{G}_{t'}),$$

$$A_t = \alpha(t) \cup \delta_t, \text{ and} \qquad B_t = X_t \setminus A_t = X_t \cap \beta(t) \setminus \delta_t.$$

Let also $\mathbf{F}_t$ be such that $G_t = \mathbf{F}_t \oplus \mathbf{H}_t$. Note that $|\mathsf{bd}(\mathbf{G}_{t'}) \setminus A_t| \leq 1$ for $t' \in \mathsf{Ch}_r(t)$.

We proceed in a bottom-up manner to compute $s_t^{\mathcal{X}} := \hat{\mathsf{p}}_{g,\mathsf{opt}}(G_t, \mathcal{X})$, for each $t \in V(T)$, for each $\mathcal{X} \in \mathcal{P}_p(\delta_t)$. Hence, given that $\delta_r = \emptyset$, it implies that $s_r^{\emptyset} = \mathsf{p}_{g,\mathsf{opt}}(G)$.

We fix $t \in V(T)$. By induction, for each $t' \in \mathsf{Ch}_r(t)$ and for each $\mathcal{X}_{t'} \in \mathcal{P}_p(\delta_{t'})$, we compute the value $s_{t'}^{\mathcal{X}_{t'}}$. Let $\mathcal{X} \in \mathcal{P}_p(\delta_t)$, let $\mathcal{Q}$ be the set of all $\mathcal{A} \in \mathcal{P}_p(A_t)$ such that $\mathcal{A} \cap \delta_t = \mathcal{X}$, and let $\mathcal{A} \in \mathcal{Q}$. Since Π' is $\mathcal{H}$-nice, there is an $\mathcal{H}$-nice reduction $(\mathbf{H}_{\mathcal{A}}, \mathcal{A}', s_{\mathcal{A}})$ of $(\mathbf{H}_t, \mathcal{A})$ with respect to Π'. Hence, $\hat{\mathsf{p}}_{g,\mathsf{opt}}(G_t, \mathcal{A}) = \hat{\mathsf{p}}_{g,\mathsf{opt}}(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}_t, \mathcal{A}') + s_{\mathcal{A}}$. Let us compute $\hat{\mathsf{p}}_{g,\mathsf{opt}}(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}_t, \mathcal{A}')$.

By definition of an $\mathcal{H}$-reduction, $(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}_t) - (\cup \mathcal{A}') \in \mathcal{H}$. Hence, we can compute $\hat{\mathsf{p}}_{g,\mathsf{opt}}(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}_t, \mathcal{A}')$, and thus $\hat{\mathsf{p}}_{g,\mathsf{opt}}(G_t, \mathcal{A})$, using Alg on the instance $(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}_t, \mathcal{A}')$. Finally, $s_t^{\mathcal{X}} = \mathsf{opt}_{\mathcal{A} \in \mathcal{Q}} \hat{\mathsf{p}}_{g,\mathsf{opt}}(G_t, \mathcal{A})$.

It remains to calculate the complexity. Throughout, we make use of the fact that p is a fixed constant. We can assume that T has at most n nodes: for any pair of nodes t and t' with $(\alpha \cup \beta)(t) \subseteq (\alpha \cup \beta)(t')$, we can contract the edge tt' of T to a new vertex t'' with $\alpha(t'') = \alpha(t')$ and $\beta(t'') = \beta(t')$. This defines a valid 1-$\mathcal{H}$-tree decomposition of the same width. For any leaf t of T, there is a vertex $u \in V(G)$ that only belongs to the bag of t. From this observation, we can inductively associate each node of T to a distinct vertex of G. Hence, if $c_t = |\mathsf{Ch}_r(t)|$, then we have $\sum_{t \in V(T)} c_t \leq n$. Let also $n_t = |(\alpha \cup \beta)(t)|$ and $m_t = |E(G[(\alpha \cup \beta)(t)])|$. Note that $|A_t| = |\alpha(t)| + |\delta_t \cap \beta(t)| \leq k+1$ and that $|B_t| = |\bigcup_{t' \in V(T)} \delta_{t'} \cap \beta(t)| \leq c_t$, so $|X_t| \leq k+1+c_t$. Moreover, the properties of the tree decompositions imply that the vertices in $\beta(t) \setminus X_t$ are only present in node t. Then, $\sum_{t \in V(T)} n_t = \sum_{t \in V(T)} (|X_t| + |\beta(t) \setminus X_t|) = \mathcal{O}(k \cdot n)$. Also, let $\bar{m}_t$ be the number of edges only present in the bag of node t. The edges that are present in several bags are those in the adhesion of t and its neighbors. t is adjacent to its $|c_t|$ children and its parent, and an adhesion has size at most $k+1$. Thus, $\sum_{t \in V(T)} m_t \leq \sum_{t \in V(T)} (\bar{m}_t + k^2(1+c_t)) = \mathcal{O}(m + k^2 \cdot n)$.

There are $p^{|A_t|} \leq p^{k+1} = \mathcal{O}(p^k)$ partitions of $\mathcal{P}_p(A_t)$. For each of them, we compute in time $\mathcal{O}(k \cdot c_t)$ an $\mathcal{H}$-nice reduction $(\mathbf{H}_{\mathcal{A}}, \mathcal{A}', s_{\mathcal{A}})$ with $| \cup \mathcal{A}'| = |A_t| + \mathcal{O}(1) = k + \mathcal{O}(1)$ and with $\mathcal{O}(|B_t|) = \mathcal{O}(c_t)$ additional vertices and edges. We thus solve Π' on $(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}_t, \mathcal{A}')$ in time $f(k + \mathcal{O}(1)) \cdot \mathcal{O}((n_t + c_t)^c \cdot (m_t + c_t)^d)$. Hence, the running time is $\mathcal{O}(p^k \cdot f(k + \mathcal{O}(1)) \cdot (k \cdot n)^c \cdot m^d)$ (or $\mathcal{O}(p^k \cdot f(k + \mathcal{O}(1)) \cdot (m + k^2 \cdot n)^d)$ if $c = 0$). $\square$

9.4.4 Generalizations

For the sake of simplicity, we assumed in Theorem 9.4.1 that the problem Π under consideration takes as input just a graph. However, a similar statement still holds if we add labels/weights on the vertices/edges of the input graph. This is in particular the case for WEIGHTED INDEPENDENT SET (Subsection 9.5.2) and MAXIMUM WEIGHTED CUT (Subsection 9.5.4) where the vertices or edges are weighted. Furthermore, while we omit the proof here, with some minor changes to the definition of a nice problem, a similar statement would also hold for q(-torso)-$\mathcal{H}$-treewidth.

Moreover, again for the sake of simplicity, we assumed that Π' is solvable in FPT-time, while other complexities such as XP-time could be considered. Similarly, in the definition of the nice reduction, the contraints $|A'| = |A| + \mathcal{O}(1)$, $|V(G')| \leq |X| + \mathcal{O}(|B|)$, $|E(G')| \leq |E(G[X])| + \mathcal{O}(|B|)$ can be modified. In those cases, the dynamic programming algorithm still holds, but the running time of Theorem 9.4.1 changes.

To give a precise running time for K_t-SUBGRAPH-COVER (Subsection 9.5.1), WEIGHTED INDEPENDENT SET (Subsection 9.5.2), and MAXIMUM WEIGHTED CUT (Subsection 9.5.4) below, let us observe that, if Π' is solvable in time $f(|\cup \mathcal{X}|) \cdot (n')^c \cdot (m')^d$, where $G' = G - \cup \mathcal{X}$, $n' = |V(G')|$, and $m' = |E(G')|$, then the running time of Theorem 9.4.1 is better. Indeed, in the proof of the complexity of Theorem 9.4.1, we now solve Π' on $(\mathbf{H}_{\mathcal{A}} \triangleright \mathbf{F}, \mathcal{A}')$ in time $f(k + \mathcal{O}(1)) \cdot \mathcal{O}((n_t' + c_t)^c \cdot (m_t' + c_t)^d)$, where $n_t' = |\beta(t)|$ and $m_t' = |E(G[\beta(t)])|$. We have

$\sum_{t\in V(T)} n'_t = \sum_{t\in V(T)}(|B| + |\beta(t) \cap \delta_t| + |\beta(t) \setminus X|) = \mathcal{O}(n)$ and $\sum_{t\in V(T)} m'_t \leq m$. Hence, the total running time is $\mathcal{O}(p^k \cdot (k \cdot n + f(k + \mathcal{O}(1)) \cdot n^c \cdot m^d))$.

9.5 Applications

We now apply the above framework to give FPT-algorithms for several problems parameterized by bipartite treewidth, that is, 1-$\mathcal{B}$-treewidth where $\mathcal{B}$ is the class of bipartite graphs. Thanks to Theorem 9.4.1, this now reverts to showing that the problem under consideration has a $\mathcal{B}$-nice annotated extension that is solvable in FPT-time when parameterized by oct. Several of the presented results actually hold for other graph classes $\mathcal{H}$, not necessarily only bipartite graphs.

More particularly, we study K_t-Subgraph-Cover, Weighted Vertex Cover, Odd Cycle Transversal, and Maximum Weighted Cut in Subsection 9.5.1, Subsection 9.5.2, Subsection 9.5.3, and Subsection 9.5.4, respectively. Given that Maximum Weighted Cut is not a graph modification problem, we decided to not include the entire proof of the results here, and we refer the reader to [171] for the full proof. Finally, in Subsection 9.5.5, we discuss the hardness of vertex deletion problems parameterized by btw.

All of the problems of this section have the following property, that seems critical to show that a problem is $\mathcal{H}$-nice.

Gluing property. Let Π be a p-annotated problem corresponding to some choice of p-partition-evaluation function f and some $\mathsf{opt} \in \{\max, \min\}$. We say that Π has the *gluing property* if, given two compatible boundaried graphs $\mathbf{F} = (F, X, \rho)$ and $\mathbf{G} = (G, X, \rho)$, $\mathcal{X} \in \mathcal{P}_p(X)$, and $\mathcal{P} \in \mathcal{P}_p(V(\mathbf{F} \oplus \mathbf{G}))$ such that $\mathcal{X} \subseteq \mathcal{P}$, then $\hat{\mathsf{p}}_{f,\mathsf{opt}}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = f(\mathbf{F} \oplus \mathbf{G}, \mathcal{P})$ if and only if $\hat{\mathsf{p}}_{f,\mathsf{opt}}(F, \mathcal{X}) = f(F, \mathcal{P} \cap V(F))$ and $\hat{\mathsf{p}}_{f,\mathsf{opt}}(G, \mathcal{X}) = f(G, \mathcal{P} \cap V(G))$, where F and G are the underlying graphs of $\mathbf{F}$ and $\mathbf{G}$ (see Figure 9.6).

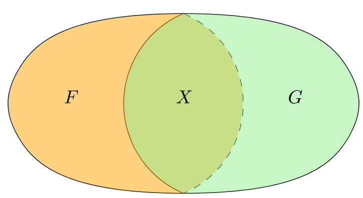

Figure 9.6: If a problem Π has the gluing property, then a solution is optimal on $\mathbf{F} \oplus \mathbf{G}$ if and only if its restriction to F and its restriction to G are both optimal.

For the sake of simplicity, with a slight abuse of notation, we identify in this section a vertex with its heir.

Let Π' be an annotated extension of some problem Π. Given an instance $(\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i), (A, B), \mathcal{A})$ for a $\mathcal{B}$-nice reduction with respect to Π', we know that the boundary of each G_i contains at most one vertex of B, and hence which is not annotated. To show that Π' is $\mathcal{B}$-nice, we thus essentially need to show how to reduce a graph $\mathbf{F} \oplus \mathbf{G}$ to a graph F' when the boundary of $\mathbf{F}$ and $\mathbf{G}$ is totally annotated (and that Π' has the gluing property), and when the boundary of $\mathbf{F}$ and $\mathbf{G}$ has a single vertex v that is not annotated. To show that Π is FPT parameterized by btw, it then suffices to prove that Π' is FPT parameterized by oct on instances where a minimal odd cycle transversal is annotated.

9.5.1 K_t-Subgraph-Cover

Let $\mathcal{H}$ be a graph class. Recall that the problem Vertex Deletion to $\mathcal{H}$ is defined as follows.

(Weighted) Vertex Deletion to $\mathcal{H}$

Input:	A graph G (and a weight function $w : V(G) \to \mathbb{N}$).
Question:	Find the set $S \subseteq V(G)$ of minimum size (resp. weight) such that $G - S \in \mathcal{H}$.

Given a graph H, if $\mathcal{H}$ is the class of graphs that do not contain H as a subgraph (resp. a minor/odd-minor/induced subgraph), then the corresponding problem is called H-Subgraph-Cover (resp. H-Minor-Cover/H-Odd-Minor-Cover/H-Induced-Subgraph-Cover).

Let H be a graph and $w : V(G) \to \mathbb{N}$ be a weight function (assigning one to every vertex in the unweighted case). We define f_H as the 2-partition-evaluation function where, for every graph G, for every $(R, S) \in \mathcal{P}_2(V(G))$,

$$f_H(G,(R,S)) = \begin{cases} +\infty & \text{if } H \text{ is a subgraph of } G - S, \\ w(S) & \text{otherwise.} \end{cases}$$

Seen as an optimization problem, (Weighted) H-Subgraph-Cover is the problem of computing $\mathsf{p}_{f_H,\min}(G)$. We call its annotated extension (Weighted) Annotated H-Subgraph-Cover. In other words, (Weighted) Annotated H-Subgraph-Cover is defined as follows.

(Weighted) Annotated H-Subgraph-Cover

Input:	A graph G, two disjoint sets $R, S \subseteq V(G)$ (and a weight function $w : V(G) \to \mathbb{N}$).
Question:	Find, if it exists, the minimum size (resp. weight) of a set $S^\star \subseteq V(G)$ such that $R \cap S^\star = \emptyset$, $S \subseteq S^\star$, and $G - S^\star$ does not contain H as a subgraph

Given a graph G and a set $X \subseteq V(G)$, note that X is a vertex cover if and only if $V(G) \setminus X$ is an independent set. Hence, the size/weight of a minimum vertex cover is equal to the size/weight of a maximum independent set. Thus, seen as optimization problems, (Weighted) Vertex Cover and (Weighted) Independent Set are equivalent problems.

In order to prove that (Weighted) Annotated K_t-Subgraph-Cover is a nice problem, we first prove that (Weighted) Annotated K_t-Subgraph-Cover has the gluing property.

Lemma 9.5.1 (Gluing property)**.** (Weighted) Annotated K_t-Subgraph-Cover *has the gluing property. More precisely, given two boundaried graphs $\mathbf{F} = (F, B_F, \rho_F)$ and $\mathbf{G} = (G, B_G, \rho_G)$, a weight function $w : V(\mathbf{F} \oplus \mathbf{G}) \to \mathbb{N}$, a set $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ such that $B_F \cap B_G \subseteq X$, and $\mathcal{X} = (R, S) \in \mathcal{P}_2(X)$, we have*

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{K_t},\min}(G, \mathcal{X} \cap V(G), w) - \bar{w},$$

where $\bar{w} = w(S \cap B_F \cap B_G)$.

Proof. Observe that K_t is a subgraph of $\mathbf{F} \oplus \mathbf{G}$ if and only if K_t is a subgraph of F or of G.

Let $\mathcal{P} = (R^\star, S^\star) \in \mathcal{P}_2(V(\mathbf{F} \oplus \mathbf{G}))$ be such that $\mathcal{X} \subseteq \mathcal{P}$ and $\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = f_{K_T}(\mathbf{F} \oplus \mathbf{G}, \mathcal{P}, w)$. Then K_t is neither a subgraph of $F - (S^\star \cap V(F))$ nor of $G - (S^\star \cap V(G))$. Therefore,

$$\begin{aligned} \hat{\mathsf{p}}_{f_{K_T},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) &= w(S^\star) \\ &= w(S^\star \cap V(F)) + w(S^\star \cap V(G)) - w(S^\star \cap B_F \cap B_G) \\ &\geq \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{K_t},\min}(G, \mathcal{X} \cap V(G), w) - \bar{w}. \end{aligned}$$

Reciprocally, let $\mathcal{P}_H = (R_H, S_H) \in \mathcal{P}_2(V(H))$ be such that $\mathcal{X} \cap V(H) \subseteq \mathcal{P}_H$ and $\hat{\mathsf{p}}_{f_{K_t},\min}(H, \mathcal{X} \cap V(H)) = f_{K_t}(H, \mathcal{P}_H)$ for $H \in \{F, G\}$. Then K_t is not a subgraph of $(\mathbf{F} \oplus \mathbf{G}) - (S_F \cup S_G)$, so

$$\begin{aligned}\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) &\leq w(S_F \cup S_G) \\ &= w(S_F) + w(S_G) - \bar{w} \\ &= \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{K_t},\min}(G, \mathcal{X} \cap V(G), w) - \bar{w}.\end{aligned}$$

□

The main obstacle to find an FPT-algorithm parameterized by $(1, \mathcal{H})$-tw for (Weighted) Annotated H-Subgraph-Cover, for H that is not a clique, is the fact that the problem does not have the gluing property.

Lemma 9.5.2. *If H is not a complete graph, then* (Weighted) Annotated H-Subgraph-Cover *does not have the gluing property.*

Proof. Since H is not a clique, there are two vertices $u, v \in V(H)$ that are not adjacent. Let $V' = V(H) \setminus \{u, v\}$ and let $\sigma : V' \to \mathbb{N}$ be an injective function. Let $\mathbf{F} = (H - u, V', \sigma)$ and $\mathbf{G} = (H - v, V', \sigma)$. Then $\mathbf{F} \oplus \mathbf{G}$ is isomorphic to H. Let $\mathcal{X} = (V', \emptyset)$ and $\mathcal{P} = (H - u, \{u\}) \supseteq \mathcal{X}$. Then we have $\hat{\mathsf{p}}_{f_H,\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = f_H(\mathbf{F} \oplus \mathbf{G}, \mathcal{P}) = 1$. However, $\hat{\mathsf{p}}_{f_H,\min}(G, \mathcal{X} \cap V(G)) = 0 < f_H(G, \mathcal{P} \cap V(G)) = 1$. □

We now show how to reduce a graph $\mathbf{F} \oplus \mathbf{G}$ when the boundary of $\mathbf{F}$ and $\mathbf{G}$ has a single vertex v that is not annotated. Essentially, given an annotation (R, S), we compare the size s^+ of an optimal solution when v is added to S and the size s^- of an optimal solution when v is added to R. If $s^+ \leq s^-$, then it is always better too add v to S, so we do so and delete $V(G) \setminus V(F)$ from $\mathbf{F} \oplus \mathbf{G}$. Otherwise, we still delete $V(G) \setminus V(F)$ from $\mathbf{F} \oplus \mathbf{G}$, but there is no need to annotate v, whose addition to S or not will be determined by its behaviour in F. See Figure 9.7 for an illustration.

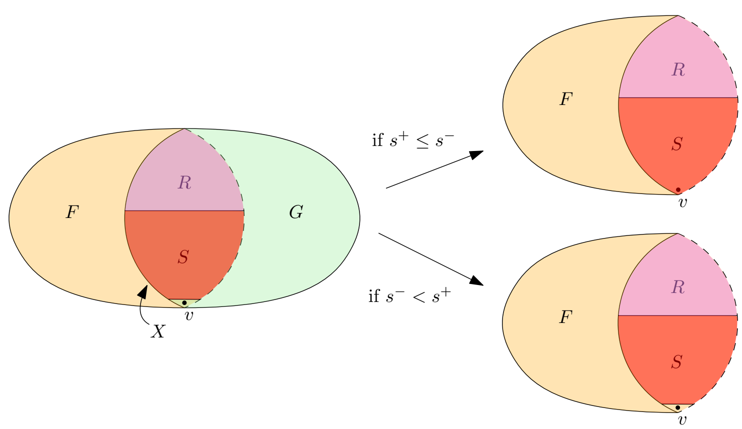

Figure 9.7: Illustration of the gadgetization for K_t-Subgraph-Cover.

Lemma 9.5.3 (Gadgetization)**.** *Let $\mathbf{F} = (F, B_F, \rho_F)$ and $\mathbf{G} = (G, B_G, \rho_G)$ be two boundaried graphs. Let $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ be such that $B_F \cap B_G \subseteq X$. Let also $v \in B_F \cap B_G$ and let $\mathcal{X} = (R, S) \in \mathcal{P}_2(X \setminus \{v\})$. We define $\mathcal{X}^+ = (R, S \cup \{v\})$ and $\mathcal{X}^- = (R \cup \{v\}, S)$. Furthermore, for $a \in \{+, -\}$, we set $s^a = \hat{\mathsf{p}}_{f_{K_t},\min}(G, \mathcal{X}^a \cap V(G))$, and we set $\bar{s} = |S \cap B_F \cap B_G|$. Then, for any $t \in \mathbb{N}$,*

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = \begin{cases} s^+ + \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X}^+ \cap V(F)) - \bar{s} - 1 & \text{if } s^+ \leq s^-, \\ s^- + \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X} \cap V(F)) - \bar{s} & \text{otherwise.} \end{cases}$$

Proof. For $H \in \{F, G\}$ and for $a \in \{+,-\}$, let $S_H^a \subseteq V(H)$ be such that $\mathcal{X}^a \cap V(H) \subseteq (V(H) \setminus S_H^a, S_H^a)$ and $\hat{\mathsf{p}}_{f_{K_t},\min}(H, \mathcal{X}^a \cap V(H)) = f_{K_t}(H, (V(H) \setminus S_H^a, S_H^a)) = |S_H^a|$. We deduce that $s^+ = |S_G^+|$ and $s^- = |S_G^-|$. Let us note similarly $t^+ = |S_F^+|$ and $t^- = |S_F^-|$.

Note that $S_F^+ \cap S_G^+ = (S \cap B_F \cap B_G) \cup \{v\}$ and $S_F^- \cap S_G^- = S \cap B_F \cap B_G$. Hence, using Lemma 9.5.1, we have that

$$\begin{aligned}\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) &= \min\{\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}^+), \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}^-)\}\\ &= \min\{t^+ + s^+ - 1, t^- + s^-\} - \bar{s}.\end{aligned}$$

Note that we always have $s^+ \leq s^- + 1$ since $G - (S_G^- \cup \{v\})$ does not contain K_t as a subgraph, and thus $|S_G^- \cup \{v\}| \geq p_{f_{K_t},\min}(G, \mathcal{X}^+ \cap V(G))$. Similarly, $t^+ \leq t^- + 1$.

Thus, if $s^+ \leq s^-$, then $t^+ + s^+ - 1 \leq t^- + s^+ \leq t^- + s^-$. Given that $t^+ = \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X}^+ \cap V(F))$, it follows that

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X}^+ \cap V(F)) + s^+ - \bar{s} - 1.$$

And if $s^- < s^+$, then $s^+ = s^- + 1$, so $\min\{t^+ + s^+ - 1, t^- + s^-\} = \min\{t^+, t^-\} + s^- = \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X} \cap V(F)) + s^-$. It follows that

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X} \cap V(F)) + s^- - \bar{s}.$$

□

Contrary to Lemma 9.5.1, observe that Lemma 9.5.3 only holds in the unweighted case. Indeed, in the weighted case, we now have $s^+ \leq s^- + w(v)$, and thus, when $s^+ \in [s^- + 1, s^- + w(v) - 1]$, we do not know what happens.

Using Lemma 9.5.1 and Lemma 9.5.3, we can now prove that Annotated K_t-Subgraph-Cover is $\mathcal{H}$-nice. Essentially, given an instance $(\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i \in [d]} \mathbf{G}_i), (A, B), (R, S))$, we reduce $\mathbf{G}$ to $\mathbf{X}$ and further remove some vertices of B that can be optimally added to S, and show that the resulting boundaried graph is equivalent to $\mathbf{G}$ modulo some constant s.

Lemma 9.5.4 (Nice problem)**.** *Let $\mathcal{H}$ be a hereditary graph class. Let $t \in \mathbb{N}$.* Annotated K_t-Subgraph-Cover *is $\mathcal{H}$-nice.*

Proof. Let $\mathbf{G} = (G, X, \rho)$ be a boundaried graph, let $\mathbf{X} = (G[X], X, \rho_X)$ be a trivial boundaried graph and let $\{\mathbf{G}_i = (G_i, X_i, \rho_i) \mid i \in [d]\}$ be a collection of boundaried graphs, such that $\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i \in [d]} \mathbf{G}_i)$, let (A, B) be a partition of X such that for all $i \in [d]$, $|X_i \setminus A| \leq 1$, and let $\mathcal{A} = (R, S) \in \mathcal{P}_2(A)$. Suppose that we know, for every $i \in [d]$ and each $\mathcal{X}_i \in \mathcal{P}_2(X_i)$, the value $\hat{\mathsf{p}}_{f_{K_t},\mathsf{min}}(G_i, \mathcal{X}_i)$.

Let $(\mathbf{H}_0, S_0, s_0) = (\mathbf{G}, S, 0)$. For i going from 1 up to d, we construct $(\mathbf{H}_i, S_i, s_i)$ from $(\mathbf{H}_{i-1}, S_{i-1}, s_{i-1})$ such that for any boundaried graph $\mathbf{F}$ compatible with $\mathbf{G}$,

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i \oplus \mathbf{F}, \mathcal{A}_i) + s_i,$$

where $\mathcal{A}_i = (R, S_i)$. This is obviously true for $i = 0$.

Let $i \in [d]$. Let $\mathbf{H}_i$ be the boundaried graph such that $\mathbf{H}_{i-1} = \mathbf{H}_i \boxplus \mathbf{G}_i$. See Figure 9.8 for an illustration. By induction, $\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{A}_{i-1}) + s_{i-1}$.

Suppose first that $X_i \subseteq R \cup S_{i-1}$. Let $\mathcal{P}_i = (R^i, S^i) \in \mathcal{P}_2(X)$ be such that $\mathcal{A}_{i-1} \subseteq \mathcal{P}_i$ and $\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{A}_{i-1}) = \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{P}_i)$. Let F be the underlying graph of $\mathbf{F}$. According

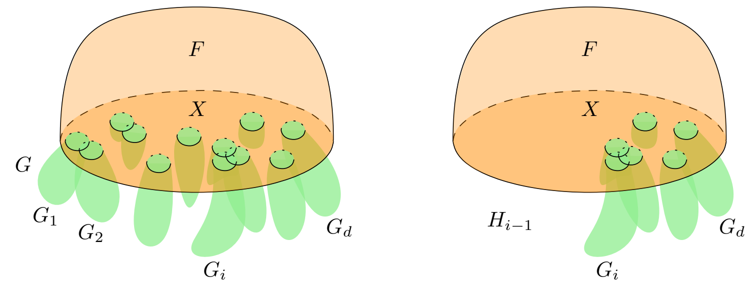

Figure 9.8: Illustration of $\mathbf{G}$ and $\mathbf{H_{i-1}}$ in the proof of Lemma 9.5.4.

to Lemma 9.5.1,

$$\begin{aligned}\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1}\oplus\mathbf{F},\mathcal{P}_i) &= \hat{\mathsf{p}}_{f_{K_t},\min}(F,\mathcal{P}_i)+\hat{\mathsf{p}}_{f_{K_t},\min}(H_{i-1},\mathcal{P}_i)-|X\cap S^i|\\ &= \hat{\mathsf{p}}_{f_{K_t},\min}(F,\mathcal{P}_i)+\hat{\mathsf{p}}_{f_{K_t},\min}(H_i,\mathcal{P}_i)\\ &\quad+\hat{\mathsf{p}}_{f_{K_t},\min}(G_i,\mathcal{P}_i\cap X_i)-|X_i\cap S^i|-|X\cap S^i|\\ &= \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i\oplus\mathbf{F},\mathcal{P}_i)+\hat{\mathsf{p}}_{f_{K_t},\min}(G_i,\mathcal{A}_{i-1}\cap X_i)-|S_{i-1}\cap X_i|.\end{aligned}$$

Since this is the case for all such $\mathcal{P}_i$, it implies that

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1}\oplus\mathbf{F},\mathcal{A}_{i-1})=\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i\oplus\mathbf{F},\mathcal{A}_{i-1})+\hat{\mathsf{p}}_{f_{K_t},\min}(G_i,\mathcal{A}_{i-1}\cap X_i)-|S_{i-1}\cap X_i|.$$

Therefore,

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{G}\oplus\mathbf{F},\mathcal{A})=\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i\oplus\mathbf{F},\mathcal{A}_i)+s_i,$$

where $\mathcal{A}_i=\mathcal{A}_{i-1}$ and $s_i=s_{i-1}+\hat{\mathsf{p}}_{f_{K_t},\min}(G_i,\mathcal{A}_{i-1}\cap X_i)-|S_{i-1}\cap X_i|$.

Otherwise, there is $v_i\in V(G_{i-1})$ such that $X_i\setminus(R\cup S_{i-1})=\{v_i\}$. Let $\mathcal{X}_i^+=(R,S_{i-1}\cup\{v_i\})\cap X_i$ and $\mathcal{X}_i^-=(R\cup\{v_i\},S_{i-1})\cap X_i$. Let $\mathcal{P}_i=(R^i,S^i)\in\mathcal{P}_2(X\setminus\{v_i\})$ be such that $\mathcal{A}_{i-1}\subseteq\mathcal{P}_i$ and $\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1}\oplus\mathbf{F},\mathcal{A}_{i-1})=\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1}\oplus\mathbf{F},\mathcal{P}_i)$. Note that

$$\mathbf{H}_{i-1}\oplus\mathbf{F}=(\mathbf{H}_i\boxplus\mathbf{G}_i)\oplus\mathbf{F}=(\mathbf{H}_i\boxplus\mathbf{F})\oplus\mathbf{G}_i.$$

For $a\in\{+,-\}$, let $s_i^a=\mathsf{p}_{f_{K_t},\min}(G,\mathcal{X}_i^a)$. Then, using Lemma 9.5.3, we have the following case distinction.

$$\begin{aligned}\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_{i-1}\oplus\mathbf{F},\mathcal{P}_i) &= \hat{\mathsf{p}}_{f_{K_t},\min}((\mathbf{H}_i\boxplus\mathbf{F})\oplus\mathbf{G}_i\oplus\mathbf{F},\mathcal{P}_i)\\ &= \begin{cases} s_i^+ + \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i\oplus\mathbf{F},(R^i,S^i\cup\{v_i\}))-|S_{i-1}\cap X_i|-1 & \text{if } s_i^+\le s_i^-\\ s_i^- + \hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i\oplus\mathbf{F},\mathcal{P}_i)-|S_{i-1}\cap X_i| & \text{otherwise.}\end{cases}\end{aligned}$$

Since this is the case for every such $\mathcal{P}_i$, by setting $(\mathbf{H}_i,S_i,s_i)=(\mathbf{H}_i,S_{i-1}\cup\{v_i\},s_{i-1}+s^+-|S_{i-1}\cap X_i|-1)$ if $s^+\le s^-$ and $(\mathbf{H}_i,S_i,s_i)=(\mathbf{H}_i,S_{i-1},s_{i-1}+s^--|S_{i-1}\cap X_i|)$ otherwise, we have that

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{G}\oplus\mathbf{F},\mathcal{A})=\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_i\oplus\mathbf{F},\mathcal{A}_i)+s_i.$$

The boundaried graph $\mathbf{H_d}$ obtained at the end is isomorphic to $\mathbf{X}$. Let $S_B=S_d\setminus S\subseteq B$. Observe that

$$\begin{aligned}\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{H}_d\oplus\mathbf{F},\mathcal{A}_d) &= \hat{\mathsf{p}}_{f_{K_t},\min}((\mathbf{H}_d\oplus\mathbf{F})-S_B,\mathcal{A}_d\setminus S_B)+|S_B|\\ &= \hat{\mathsf{p}}_{f_{K_t},\min}((\mathbf{H}_d-S_B)\triangleright\mathbf{F},\mathcal{A})+|S_B|\end{aligned}$$

Hence,

$$\hat{\mathsf{p}}_{f_{K_t},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{K_t},\min}((\mathbf{X} - S_B) \triangleright \mathbf{F}, \mathcal{A}) + |S_B| + s_d.$$

Let $\mathbf{H} = \mathbf{X} - S_B$. Note that $|V(H)| \leq |X|$ and $|E(H)| \leq E(G[X])$, where H is the underlying graph of $\mathbf{H}$. Given that $S_B \subseteq B \subseteq X$, it implies that $\mathbf{H} \triangleright \mathbf{F}$ is isomorphic to $F - S_B$. Thus, since $\mathcal{H}$ is hereditary, if $F - A$ belongs to $\mathcal{H}$, then so does $(\mathbf{H} \triangleright \mathbf{F}) - A$. Hence, $(\mathbf{X} - S_B, \mathcal{A}, |S_B| + s_d)$ follows every conditions so that it is an $\mathcal{H}$-nice reduction of $(G, \mathcal{A})$ with respect to Annotated K_t-Subgraph-Cover.

At each step i, we compute $|S_{i-1} \cap X_i|$, and thus s_i, in time $\mathcal{O}(|A|)$ (since $\hat{\mathsf{p}}_{f_{K_t},\mathsf{min}}(G_i, \mathcal{X}_i)$ is supposed to be known). $\mathbf{H}_i$ and S_i are then constructed in time $\mathcal{O}(1)$. Hence, the computation takes time $\mathcal{O}(|A| \cdot d)$, and thus, Annotated K_t-Subgraph-Cover is $\mathcal{H}$-nice. □

We now solve Annotated K_t-Subgraph-Cover for $t \geq 3$ parameterized by oct. Note that Vertex Cover can be solved on bipartite graphs in time $\mathcal{O}(m\sqrt{n})$ using a maximum matching algorithm [231] due to Kőnig's theorem [87]. Moreover, Weighted Vertex Cover can be solved on bipartite graphs in time $\mathcal{O}(m \cdot n)$ using a flow algorithm [198, 246]. Indeed, let $G = (A, B)$ be a bipartite graph and $w : V(G) \to \mathbb{N}$ be a weight function. We construct a flow network N by connecting a source s to each vertex in A and a sink t to each vertex in B. We give infinite capacity to the original edges of G, and capacity $w(v)$ to each edge connecting a vertex v and a terminal vertex. Every $s - t$ cut in N corresponds to exactly one vertex cover and every vertex cover corresponds to an $s - t$ cut. Thus a minimum cut of N gives a minimum weight vertex cover of G.

Lemma 9.5.5. *Let $t \in \mathbb{N}_{\geq 3}$. There is an algorithm that, given a graph G and two disjoint sets $R, S \subseteq V(G)$ such that $G' = G - (R \cup S)$ is bipartite, solves* Annotated K_t-Subgraph-Cover *(resp.* Weighted Annotated K_t-Subgraph-Cover*) on (G, R, S) in time $\mathcal{O}(k^t \cdot (n' + m') + m'\sqrt{n'})$ (resp. $\mathcal{O}(k^t \cdot (n' + m') + m' \cdot n')$), where $k = |R|$, $n' = |V(G')|$, and $m' = |E(G')|$.*

Proof. Observe that we can assume that $S = \emptyset$, since $S^\star$ is an optimal solution for (G, R, S) if and only if $S^\star \setminus S$ is an optimal solution for $(G - S, R, \emptyset)$. Thus, $G - R$ is bipartite, so for any occurrence of K_t contained in G (as a subgraph), at most two of its vertices belong to $G - R$. Hence, enumerating the occurrences of K_t takes time $\mathcal{O}(k^t + k^{t-1} \cdot n' + k^{t-2} \cdot m')$. If $G[R]$ contains an occurrence of K_t, then Annotated K_t-Subgraph-Cover has no solution. Let us thus assume that $G[R]$ contains no K_t. For each occurrences of K_t in G that contains $t - 1$ vertices of R and one vertex $v \in V(G) \setminus R$, we add v to $S^\star$ and remove v from G, since v has to be taken in the solution. Hence, all that remains are occurrences of K_t with $t - 2$ vertices in R and the two others in $G - R$. Let H be the graph induced by the edges of the occurrences of K_t in G with both endpoints in $G - R$. Each edge of H intersects any solution $\bar{S}$ on G for Annotated K_t-Subgraph-Cover. Hence, $\bar{S}$ is the union of $S^\star$ and a minimum (weighted) vertex cover C of H. Thus, C can be computed in time $\mathcal{O}(m'\sqrt{n'})$ (resp. $\mathcal{O}(m' \cdot n')$). The running time of the algorithm is hence $\mathcal{O}(k^t + k^{t-1} \cdot n' + k^{t-2} \cdot m' + m'\sqrt{n'})$ (resp. $\mathcal{O}(k^t + k^{t-1} \cdot n' + k^{t-2} \cdot m' + m' \cdot n')$). □

We apply Lemma 9.5.4 and Lemma 9.5.5 to the dynamic programming algorithm of Theorem 9.4.1 to obtain the following result.

Corollary 9.5.6. *Let $t \in \mathbb{N}_{\geq 3}$. Given a graph G and a bipartite tree decomposition of G of width k, there is an algorithm that solves* K_t-Subgraph-Cover *on G in time $\mathcal{O}(2^k \cdot (k^t \cdot (n + m) + m\sqrt{n}))$.*

We find a better running time when $t = 2$, i.e., for Vertex Cover/ Independent Set.

Observation 9.5.7. *Let $\mathcal{H}$ be a hereditary graph class such that* (Weighted) Vertex Cover *can be solved on instances (G, w) where $G \in \mathcal{H}$ in time $\mathcal{O}(n^c \cdot m^d)$ for some $c, d \in \mathbb{N}$. Then* (Weighted)

Annotated Vertex Cover *is solvable on instance* (G, R, S, w) *such that* $G' = G - (R \cup S) \in \mathcal{H}$ *in time* $\mathcal{O}(k \cdot (k + n') + n'^c \cdot m'^d)$, *where* $n' = |V(G')|$, $m' = |E(G')|$, *and* $k = |R|$.

Proof. Let G be a graph, w be a weight function, and $R, S \subseteq V(G)$ be two disjoint sets such that $G - (R \cup S) \in \mathcal{H}$. If R is not an independent set, then (G, R, S, w) has no solution. Hence, we assume that R is an independent set. Then $S^\star \subseteq V(G)$ is a solution of Weighted Annotated Vertex Cover on (G, R, S, w) if and only if $S^\star = S_B \cup S \cup N_G(R)$ where S_B is a solution of Weighted Vertex Cover on $(G - (R \cup S), w)$. Checking that R is an independent set takes time $\mathcal{O}(k^2)$ and then finding $N_G(R)$ takes time $\mathcal{O}(k \cdot n')$, hence the result. $\square$

We apply Lemma 9.5.4 and Observation 9.5.7 to the dynamic programming algorithm of Theorem 9.4.1 to obtain the following result.

Corollary 9.5.8. *Let* $\mathcal{H}$ *be a hereditary graph class. Suppose that* Vertex Cover /Independent Set *can be solved on* $\mathcal{H}$ *in time* $\mathcal{O}(n^c \cdot m^d)$. *Then, given a graph* G *and a* 1-$\mathcal{H}$*-tree decomposition of* G *of width* k, *there is an algorithm that solves* Vertex Cover/Independent Set *on* G *in time* $\mathcal{O}(2^k \cdot (k \cdot (k + n) + n^c \cdot m^d))$.

As a corollary of Corollary 9.5.8, we obtain the following result concerning bipartite treewidth.

Corollary 9.5.9. *Given a graph* G *and a bipartite tree decomposition of* G *of width* k, *there is an algorithm that solves* Vertex Cover/Independent Set *on* G *in time* $\mathcal{O}(2^k \cdot (k \cdot (k + n) + m\sqrt{n}))$.

9.5.2 Weighted Vertex Cover/Weighted Independent Set

Given that Lemma 9.5.3 only holds for K_t-Subgraph-Cover in the unweighted case, we propose here an analogous result that holds in the weighted case, when we restrict ourselves to $t = 2$, i.e., Weighted Vertex Cover. We already know that Weighted Vertex Cover has the gluing property (Lemma 9.5.1). We now show how to reduce a graph $\mathbf{F} \oplus \mathbf{G}$ to a graph F' when the boundary of $\mathbf{F}$ and $\mathbf{G}$ has a single vertex v that is not annotated. Recall that this reduction was sketched in Section 9.1, and see Figure 9.9 for an illustration.

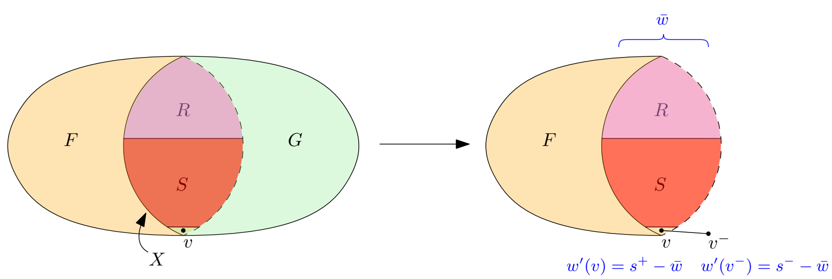

Figure 9.9: Illustration of the gadgetization for Weighted Vertex Cover.

Lemma 9.5.10 (Gadgetization)**.** *Let* $\mathbf{F} = (F, B_F, \rho_F)$ *and* $\mathbf{G} = (G, B_G, \rho_G)$ *be two boundaried graphs. Let* $w : V(\mathbf{F} \oplus \mathbf{G}) \to \mathbb{N}$ *be a weight function, let* $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ *be such that* $B_F \cap B_G \subseteq X$, *let* $v \in B_F \cap B_G$, *and let* $\mathcal{X} = (R, S) \in \mathcal{P}_2(X \setminus \{v\})$. *We define* $\mathcal{X}^+ = (R, S \cup \{v\})$ *and* $\mathcal{X}^- = (R \cup \{v\}, S)$. *Furthermore, for* $a \in \{+, -\}$, *we set* $s^a = \hat{\mathsf{p}}_{f_{K_t},\min}(G, \mathcal{X}^a \cap V(G), w)$. *We also set* $\bar{w} = w(S \cap B_F \cap B_G)$, $\mathbf{G}' = (G', \{v\}, \rho|_{\{v\}})$, *where* G' *is an edge* vv^- *for some new vertex* v^-, *and* $w' : V(\mathbf{F} \oplus \mathbf{G}') \to \mathbb{N}$ *such that* $w'(v) = s^+ - \bar{w}$, $w'(v^-) = s^- - \bar{w}$, *and* $w'(x) = w(x)$ *otherwise. Then*

$$\hat{p}_{fK_2,\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = \hat{p}_{fK_2,\min}(\mathbf{F} \oplus \mathbf{G}', \mathcal{X}, w').$$

Proof. For $a \in \{+,-\}$, let $t^a = \hat{\mathsf{p}}_{f_{K_t},\min}(F, \mathcal{X}^a \cap V(F), w)$.

Note that

$$\begin{aligned}
s'^- := \hat{\mathsf{p}}_{f_{K_2},\min}(G', \mathcal{X}^- \cap V(G'), w') &= \hat{\mathsf{p}}_{f_{K_2},\min}(G', (\{v\}, \emptyset), w')\\
&= w'(v^-)\\
&= s^- - \bar{w},\\
s'^+ := \hat{\mathsf{p}}_{f_{K_2},\min}(G', \mathcal{X}^+ \cap V(G'), w') &= \hat{\mathsf{p}}_{f_{K_2},\min}(G', (\emptyset, \{v\}), w')\\
&= w'(v)\\
&= s^+ - \bar{w},\\
t'^- := \hat{\mathsf{p}}_{f_{K_2},\min}(F, \mathcal{X}^- \cap V(F), w') &= t^-, \text{ and}\\
t'^+ := \hat{\mathsf{p}}_{f_{K_2},\min}(F, \mathcal{X}^+ \cap V(F), w') &= t^+ + w'(v) - w(v).
\end{aligned}$$

Hence, using Lemma 9.5.1, we have that

$$\begin{aligned}
&\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w)\\
&\qquad = \min\{\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}^+, w), \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}^-, w)\}\\
&\qquad = \min\{t^+ + s^+ - w(v), t^- + s^-\} - \bar{w}\\
&\qquad = \min\{t^+ + s'^+ - w(v), t^- + s'^-\}\\
&\qquad = \min\{t'^+ + s'^+ - w'(v), t'^- + s'^-\}\\
&\qquad = \min\{\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{F} \oplus \mathbf{G}', \mathcal{X}^+, w'), \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{F} \oplus \mathbf{G}', \mathcal{X}^-, w')\}\\
&\qquad = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{F} \oplus \mathbf{G}', \mathcal{X}, w').
\end{aligned}$$

□

Using Lemma 9.5.1 and Lemma 9.5.10, we can now prove that Weighted Annotated Vertex Cover is $\mathcal{H}$-nice. Essentially, given an instance $(\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i), (A, B), (R, S), w)$, we reduce $\mathbf{G}$ to $\mathbf{X}$ where we glue an edge to some vertices in B. We then show that if the appropriate weight is given to each new vertex, then the resulting boundaried graph is equivalent to $\mathbf{G}$ modulo some constant s.

Lemma 9.5.11 (Nice problem)**.** *Let $\mathcal{H}$ be a graph class that is closed under 1-clique-sums and contains edges. Then* Weighted Annotated Vertex Cover *is $\mathcal{H}$-nice.*

Proof. Let $\mathbf{G} = (G, X, \rho)$ be a boundaried graph, let $w : V(G) \to \mathbb{N}$ be a weight function, let $\mathbf{X} = (G[X], X, \rho_X)$ be a trivial boundaried graph and let $\{\mathbf{G}_i = (G_i, X_i, \rho_i) \mid i \in [d]\}$ be a collection of boundaried graphs, such that $\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i)$, let (A, B) be a partition of X such that for all $i \in [d]$, $|X_i \setminus A| \le 1$, and let $\mathcal{A} = (R, S) \in \mathcal{P}_2(A)$. Suppose that we know, for every $i \in [d]$ and each $\mathcal{X}_i \in \mathcal{P}_2(X_i)$, the value $\hat{\mathsf{p}}_{f_{K_t},\mathsf{min}}(G_i, \mathcal{X}_i, w)$.

Let $v_1, \ldots, v_{|B|}$ be the vertices of B. For $i \in [|B|]$, let $I_i = \{j \in [d] \mid X_j \setminus A = \{v_i\}\}$. Let $I_0 = \{j \in [d] \mid X_j \subseteq A\}$. Obviously, $(I_i)_{i\in[0,|B|]}$ is a partition of $[d]$. Let $\mathbf{G}'_i = \boxplus_{j\in I_i} \mathbf{G}_j$. See Figure 9.10 for an illustration.

Let $(\mathbf{H}_{-1}, w_{-1}, s_{-1}) = (\mathbf{G}, w, 0)$. For i going from 0 up to $|B|$, we will construct $(\mathbf{H}_i, w_i, s_i)$ from $(\mathbf{H}_{i-1}, w_{i-1}, s_{i-1})$ such that $w_i|_{V(F)\setminus\{v_j | j\le i\}} = w|_{V(F)\setminus\{v_j|j\le i\}}$ and $w_i|_{V(G'_j)} = w|_{V(G'_j)}$, for $j > i$, and for any boundaried graph $\mathbf{F}$ with underlying graph F and compatible with $\mathbf{G}$,

$$\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}, w) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_i \oplus \mathbf{F}, \mathcal{A}, w_i) + s_i.$$

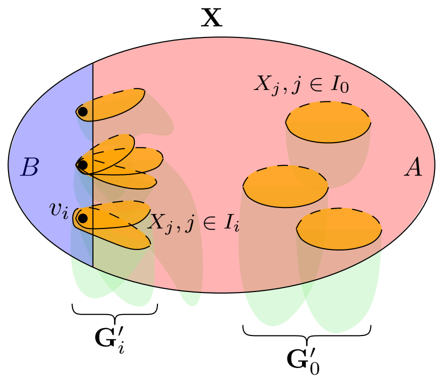

Figure 9.10: Illustration of G'_i in the proof of Lemma 9.5.11.

This is obviously true for $i = -1$.

Let $i \in [0, |B|]$. Let $\mathbf{H}'_i$ be the boundaried graph with underlying graph H'_i such that $\mathbf{H}_{i-1} = \mathbf{H}'_i \boxplus \mathbf{G}'_i$. By induction, we have

$$\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}, w) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{A}, w_{i-1}) + s_{i-1}.$$

Suppose first that $i = 0$. Let $\mathcal{P}_0 = (R^0, S^0) \in \mathcal{P}_2(X)$ be such that $\mathcal{A} \subseteq \mathcal{P}_0$ and $\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}, w) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{P}_0, w)$. According to Lemma 9.5.1,

$$\begin{aligned}
\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{P}_0, w) &= \hat{\mathsf{p}}_{f_{K_2},\min}(F, \mathcal{P}_0, w) + \hat{\mathsf{p}}_{f_{K_2},\min}(G, \mathcal{P}_0, w) - w(X \cap S^0) \\
&= \hat{\mathsf{p}}_{f_{K_2},\min}(F, \mathcal{P}_0, w) - w(X \cap S^0) + \hat{\mathsf{p}}_{f_{K_2},\min}(H'_0, \mathcal{P}_0, w) \\
&\quad + \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{K_2},\min}(G_j, \mathcal{P}_0 \cap X_j, w) - w(S^0 \cap X_j)) \\
&= \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}'_0 \oplus \mathbf{F}, \mathcal{P}_0, w) + \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{K_2},\min}(G_j, \mathcal{A} \cap X_j, w) - w(S \cap X_j)).
\end{aligned}$$

Since this is the case for all such $\mathcal{P}_0$, it implies that

$$\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}, w) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}'_0 \oplus \mathbf{F}, \mathcal{A}, w) + \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{K_2},\min}(G_j, \mathcal{A} \cap X_j, w) - w(S \cap X_j)).$$

Therefore, if $\mathbf{H}_0 = \mathbf{H}'_0$, $w_0 = w$ and

$$s_0 = \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{K_2},\min}(G_j, \mathcal{A} \cap X_j, w) - w(S \cap X_j)),$$

then

$$\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}, w) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_0 \oplus \mathbf{F}, \mathcal{A}, w_0) + s_0.$$

Otherwise, $i \in [|B|]$ and $X_j \setminus A = \{v_i\}$ for each $j \in I_i$. Let $X'_i = \bigcup_{j \in I_i} X_j$. Let $\mathcal{X}^+_i = (R, S \cup \{v_i\}) \cap X'_i$ and $\mathcal{X}^-_i = (R \cup \{v_i\}, S) \cap X'_i$. Let $\mathbf{H}''_i = (H''_i, \{v_i\}, \rho|_{\{v_i\}})$ be the boundaried graph where H''_i is an edge $v_i v^-_i$ for some new vertex v^-_i. Let $\mathbf{H}_i = \mathbf{H}'_i \boxplus \mathbf{H}''_i$. Let $s^+_i = \hat{\mathsf{p}}_{f_{K_t},\min}(G'_i, \mathcal{X}^+_i, w)$ and $s^-_i = \hat{\mathsf{p}}_{f_{K_t},\min}(G'_i, \mathcal{X}^-_i, w)$. By Lemma 9.5.1,

$$s^+_i = \sum_{j \in I_i} (\hat{\mathsf{p}}_{f_{K_t},\min}(G_j, \mathcal{X}^+_i \cap X_j, w) - w(S \cap X_j) - w(v_i)) + w(S \cap X'_i) + w(v_i),$$

and

$$s_i^- = \sum_{j \in I_i} (\hat{\mathsf{p}}_{f_{K_t},\min}(G_j, \mathcal{X}_i^- \cap X_j, w) - w(S \cap X_j)) + w(S \cap X_i').$$

Since the $\hat{\mathsf{p}}_{f_{K_t},\min}(G_j, \mathcal{X}_i^a \cap X_j, w)$ are given, s_i^+ and s_i^- can be computed in time $\mathcal{O}(|A| \cdot |I_i|)$. Let $w_i : V(\mathbf{F} \oplus \mathbf{H}_i'') \to \mathbb{N}$ be such that $w_i(v_i) = s_i^+ - w(S \cap B_F \cap B_G)$, $w_i(v_i^-) = s_i^- - w(S \cap B_F \cap B_G)$, and $w_i(x) = w(x)$ otherwise. Let $\mathcal{P}_i = (R^i, S^i) \subseteq \mathcal{P}_2(X \setminus \{v_i\})$ be such that $\mathcal{A} \subseteq \mathcal{P}_i$ and $\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{A}, w_{i-1}) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{P}_i, w_{i-1})$. Then, using Lemma 9.5.10,

$$\begin{aligned}
\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{P}_i, w_{i-1}) &= \hat{\mathsf{p}}_{f_{K_2},\min}((\mathbf{H}_i' \boxplus \mathbf{F}) \oplus \mathbf{G}_i', \mathcal{P}_i, w_{i-1}) \\
&= \hat{\mathsf{p}}_{f_{K_2},\min}((\mathbf{H}_i' \boxplus \mathbf{F}) \oplus \mathbf{H}_i'', \mathcal{P}_i, w_i) \\
&= \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_i \oplus \mathbf{F}, \mathcal{P}_i, w_i)
\end{aligned}$$

Since this is the case for all such $\mathcal{P}_i$, it implies that

$$\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}, \mathcal{A}, w_{i-1}) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_i \oplus \mathbf{F}, \mathcal{A}, w_i).$$

Therefore, given $s_i = s_{i-1}$,

$$\hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}, w) = \hat{\mathsf{p}}_{f_{K_2},\min}(\mathbf{H}_i \oplus \mathbf{F}, \mathcal{A}, w_i) + s_i.$$

Observe that $\mathbf{H}_{|\mathbf{B}|} = \mathbf{X} \boxplus (\boxplus_{i \in [|B|]} \mathbf{H}_i'')$ and $\mathbf{H}_{|\mathbf{B}|} \oplus \mathbf{F} = \mathbf{F} \oplus (\boxplus_{i \in [|B|]} \mathbf{H}_i'')$. Suppose that $F - A \in \mathcal{H}$. Given that $\mathcal{H}$ is closed under 1-clique-sums and contains edges, that each H_i'' is an edge, and that $|\mathsf{bd}(\mathbf{H}_i'')| = 1$, it follows that $(\mathbf{H}_{|\mathbf{B}|} \oplus \mathbf{F}) - A \in \mathcal{H}$. Moreover, $|V(H_{|B|})| = |X| + |B|$, and $|E(H_{|B|})| = |E(G[X])| + |B|$. Hence, $(\mathbf{H}_{|B|}, \mathcal{A}, s_{|B|}, w_{|B|})$ is an $\mathcal{H}$-nice reduction of $(\mathbf{G}, \mathcal{A}, w)$ with respect to Weighted Annotated Vertex Cover.

At each step i, s_i is computable in time $\mathcal{O}(|A| \cdot |I_i|)$, and $\mathbf{H}_i$ and w_i are computable in time $\mathcal{O}(1)$. Hence, the computation takes time $\mathcal{O}(|A| \cdot d)$. Therefore, Weighted Annotated Vertex Cover is $\mathcal{H}$-nice. □

We apply Lemma 9.5.11 and Observation 9.5.7 to the dynamic programming algorithm of Theorem 9.4.1 to obtain the following result.

Corollary 9.5.12. *Let $\mathcal{H}$ be a graph class that is closed under 1-clique-sum and contains edges. Suppose that* Weighted Vertex Cover *can be solved on instances (G, w) where $G \in \mathcal{H}$ in time $\mathcal{O}(n^c \cdot m^d)$. Then, given a graph G, a 1-$\mathcal{H}$-tree decomposition of G of width k, and a weight function w, there is an algorithm that solves* Weighted Vertex Cover/Weighted Independent Set *on (G, w) in time $\mathcal{O}(2^k \cdot (k \cdot (k + n) + n^c \cdot m^d))$.*

Given that the class $\mathcal{B}$ of bipartite graphs is closed under 1-clique-sums, that $P_2 \in \mathcal{B}$, and that Weighted Vertex Cover can be solved on bipartite graphs in time $\mathcal{O}(m \cdot n)$ [198, 246], we obtain the following result concerning bipartite treewidth using Corollary 9.5.12.

Corollary 9.5.13. *Given a graph G, a bipartite tree decomposition of G of width k, and a weight function w, there is an algorithm that solves* Weighted Vertex Cover/Weighted Independent Set *on (G, w) in time $\mathcal{O}(2^k \cdot (k \cdot (k + n) + n \cdot m))$.*

9.5.3 Odd Cycle Transversal

Let H be a graph. We define f_{oct} as the 3-partition-evaluation function where, for every graph G and for every $(S, X_1, X_2) \in \mathcal{P}_3(V(G))$,

$$f_{\mathsf{oct}}(G,(S,X_1,X_2)) = \begin{cases} |S| & \text{if } G - S \in \mathcal{B}, \text{with bipartition } (X_1,X_2), \\ +\infty & \text{otherwise.} \end{cases}$$

Hence, seen as an optimization problem, Odd Cycle Transversal is the problem of computing $\mathsf{p}_{f_{\mathsf{oct}},\min}(G)$. We call its annotated extension Annotated Odd Cycle Transversal. In other words, Annotated Odd Cycle Transversal is defined as follows.

(Weighted) Annotated Odd Cycle Transversal

Input:	A graph G, three disjoint sets $S, X_1, X_2 \subseteq V(G)$ (and a weight function $w : V(G) \to \mathbb{N}$).
Question:	Find, if it exists, a set $S^\star$ of minimum size (resp. weight) such that $S \subseteq S^\star$, $(X_1 \cup X_2) \cap S^\star = \emptyset$, and $G - S^\star$ is bipartite with X_1 and X_2 on different sides of the bipartition.

We first prove that (Weighted) Annotated Odd Cycle Transversal has the gluing property.

Lemma 9.5.14 (Gluing property)**.** (Weighted) Annotated Odd Cycle Transversal *has the gluing property. More precisely, given two boundaried graphs* $\mathbf{F} = (F, B_F, \rho_F)$ *and* $\mathbf{G} = (G, B_G, \rho_G)$, *a function* $w : V(\mathbf{F} \oplus \mathbf{G}) \to \mathbb{N}$, *a set* $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ *such that* $B_F \cap B_G \subseteq X$, *and* $\mathcal{X} = (S, X_1, X_2) \in \mathcal{P}_3(X)$, *we have*

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G, \mathcal{X} \cap V(G), w) - \bar{w},$$

where $\bar{w} = w(S \cap B_F \cap B_G)$.

Proof. Let $\mathcal{P} = (S^\star, X_1^\star, X_2^\star) \in \mathcal{P}_3(V(\mathbf{F} \oplus \mathbf{G}))$ be such that $\mathcal{X} \subseteq \mathcal{P}$ and $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = f_{\mathsf{oct}}(\mathbf{F} \oplus \mathbf{G}, \mathcal{P}, w)$. Then, for $H \in \{F, G\}$, $H - (S^\star \cap V(H))$ is bipartite, witnessed by the 2-partition $(X_1^\star \cap V(H), X_2^\star \cap V(H))$. Therefore,

$$\begin{aligned} \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) &= w(S^\star) \\ &= w(S^\star \cap V(F)) + w(S^\star \cap V(G)) - w(S^\star \cap B_F \cap B_G) \\ &\geq \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G, \mathcal{X} \cap V(G), w) - \bar{w}. \end{aligned}$$

Reciprocally, let $\mathcal{P}_H = (S_H, X_1^H, X_2^H) \in \mathcal{P}_3(V(H))$ be such that $\mathcal{X} \cap V(H) \subseteq \mathcal{P}_H$ and $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(H, \mathcal{X} \cap V(H), w) = f_{\mathsf{oct}}(H, \mathcal{P}_H, w)$ for $H \in \{F, G\}$. Since $\mathcal{P}_F \cap B_F \cap B_G = \mathcal{P}_G \cap B_F \cap B_G$, it follows that $X_1^F \cup X_1^G$ and $X_2^F \cup X_2^G$ are two independent sets of $(\mathbf{F} \oplus \mathbf{G}) - (S_F \cup S_G)$. Therefore, $(\mathbf{F} \oplus \mathbf{G}) - (S_F \cup S_G)$ is a bipartite graph witnessed by $(X_1^F \cup X_1^G, X_2^F \cup X_2^G)$. Thus,

$$\begin{aligned} \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) &\leq w(S_F \cup S_G) \\ &= w(S_F) + w(S_G) - \bar{w} \\ &= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G, \mathcal{X} \cap V(G), w) - \bar{w}. \end{aligned}$$

∎

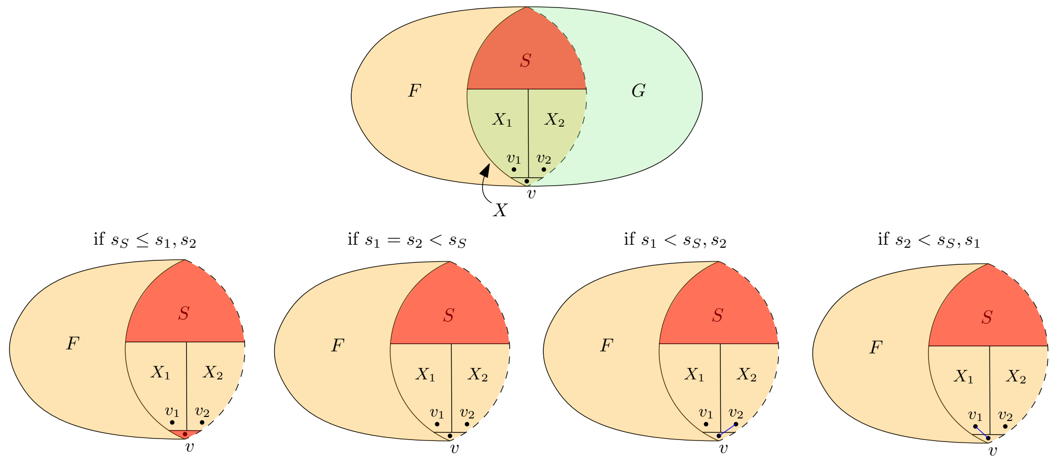

Figure 9.11: Illustration of the gadgetization for Odd Cycle Transversal.

We now show how to reduce a graph $\mathbf{F} \oplus \mathbf{G}$ to a graph F' when the boundary of $\mathbf{F}$ and $\mathbf{G}$ has a single vertex v that is not annotated. See Figure 9.11 for an illustration. Similarly to Lemma 9.5.3, the proof of Lemma 9.5.15 only holds in the unweighted case.

Lemma 9.5.15 (Gadgetization)**.** *Let $\mathbf{F} = (F, B_F, \rho_F)$ and $\mathbf{G} = (G, B_G, \rho_G)$ be two boundaried graphs. Let $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ be such that $B_F \cap B_G \subseteq X$. Let also $v \in B_F \cap B_G$, let $\mathcal{X} = (S, X_1, X_2) \in \mathcal{P}_3(X \setminus \{v\})$ with $X_1, X_2 \neq \emptyset$, and let $v_1 \in X_1$ and $v_2 \in X_2$. We set $\mathcal{X}_S = (S \cup \{v\}, X_1, X_2)$, $\mathcal{X}_1 = (S, X_1 \cup \{v\}, X_2)$, and $\mathcal{X}_2 = (S, X_1, X_2 \cup \{v\})$. We also set, for $a \in \{S, 1, 2\}$, $s_a = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G, \mathcal{X}_a \cap V(G))$, and we set $\bar{s} = |S \cap B_F \cap B_G|$. For $i \in [2]$, we note F_i the graph obtained from F by adding the edge vv_i. Then we have the following case distinction.*

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = \begin{cases} s_S + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X}_S \cap V(F)) - \bar{s} - 1 & \textit{if } s_S \leq s_1, s_2, \\ s_1 + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X} \cap V(F)) - \bar{s} & \textit{if } s_1 = s_2 < s_S, \\ s_1 + \hat{p}_{f_{\mathsf{oct}},\min}(F_2, \mathcal{X} \cap V(F)) - \bar{s} & \textit{if } s_1 < s_S, s_2, \\ s_2 + \hat{p}_{f_{\mathsf{oct}},\min}(F_1, \mathcal{X} \cap V(F)) - \bar{s} & \textit{otherwise.} \end{cases}$$

Proof. For $H \in \{F, G\}$ and $a \in \{S, 1, 2\}$, let $\mathcal{P}_H^a = (S_H^a, X_{1,H}^a, X_{2,H}^a)$ be a partition of $V(H)$ such that $\mathcal{X}_a \cap V(H) \subseteq \mathcal{P}_H^a$ and $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(H, \mathcal{X}_a \cap V(H)) = f_{\mathsf{oct}}(H, \mathcal{P}_H^a) = |S_H^a|$. We therefore have $s_a = |S_G^a|$. Let us similarly define $t_a = |S_F^a|$ for $a \in \{S, 1, 2\}$.

Note that $S_F^S \cap S_G^S = (S \cap B_F \cap B_G) \cup \{v\}$ and for $a \in \{1, 2\}$, $S_F^a \cap S_G^a = S \cap B_F \cap B_G$. Hence, using Lemma 9.5.1, we have that

$$\begin{aligned} \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) &= \min\{\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}_a) \mid a \in \{S, 1, 2\}\} \\ &= \min\{t_S + s_S - 1, t_1 + s_1, t_2 + s_2\} - \bar{s}. \end{aligned}$$

Note that $s_S \leq s_1 + 1$, since $G - (S_G^1 \cup \{v\})$ is bipartite, witnessed by the 2-partition $(X_1 \setminus \{v\}, X_2)$, and thus $|S_G^1 \cup \{v\}| \geq \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G, \mathcal{X}_S \cap V(G))$. Similarly, $s_S \leq s_2 + 1$, $t_S \leq t_1 + 1$, and $t_S \leq t_2 + 1$.

Hence, if $s_S \leq s_1, s_2$, then $\min\{t_S + s_S - 1, t_1 + s_1, t_2 + s_2\} = t_S + s_S - 1$, so

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = s_S + t_S - \bar{s} - 1.$$

If $s_1 = s_2 < s_S$, then $s_1 = s_2 = s_S - 1$. Thus, $\min\{t_S + s_S - 1, t_1 + s_1, t_2 + s_2\} = \min\{t_S, t_1, t_2\} + s_1 = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X} \cap V(F)) + s_1$, so

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = s_1 + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X} \cap V(F)) - \bar{s}.$$

If $s_1 < s_S, s_2$, then $s_1 + 1 = s_S \leq s_2$. We have $t_2 + s_2 \geq t_S + s_2 \geq t_S + s_S - 1$. Thus, $\min\{t_S + s_S - 1, t_1 + s_1, t_2 + s_2\} = \min\{t_S, t_1\} + s_1 = s_1 + \min\{\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X}_S \cap V(F)), \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{X}_1 \cap V(F))\}$. Hence, we just need to ensure that v cannot be added to X_2, which is done by adding an edge between v and $v_2 \in X_2$. Therefore,

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = s_1 + \hat{p}_{f_{\mathsf{oct}},\min}(F_2, \mathcal{X} \cap V(F)) - \bar{s}.$$

Otherwise, $s_2 < s_S, s_1$ By symmetry, we similarly obtain

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}) = s_2 + \hat{p}_{f_{\mathsf{oct}},\min}(F_1, \mathcal{X} \cap V(F)) - \bar{s}.$$

□

Using Lemma 9.5.14 and Lemma 9.5.15, we can now prove that Annotated Odd Cycle Transversal is $\mathcal{H}$-nice. Essentially, given an instance $(\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i), (A, B), \mathcal{A})$, we reduce $\mathbf{G}$ to $\mathbf{X}$ where we add two new vertices u_1 and u_2 in $\mathcal{A}$ and add edges between u_i and some vertices in B, for $i \in [2]$, and show that the resulting boundaried graph is equivalent to $\mathbf{G}$ modulo some constant s.

Lemma 9.5.16 (Nice problem)**.** *Let $\mathcal{H}$ be a hereditary graph class.* Annotated Odd Cycle Transversal *is $\mathcal{H}$-nice.*

Proof. Let $\mathbf{G} = (G, X, \rho)$ be a boundaried graph, let $\mathbf{X} = (G[X], X, \rho_X)$ be a trivial boundaried graph and let $\{\mathbf{G}_i = (G_i, X_i, \rho_i) \mid i \in [d]\}$ be a collection of boundaried graphs, such that $\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i)$, let (A, B) be a partition of X such that for all $i \in [d]$, $|X_i \setminus A| \leq 1$, and let $\mathcal{A} = (S, X^1, X^2) \in \mathcal{P}_3(A)$. Suppose that we know, for every $i \in [d]$ and each $\mathcal{X}_i \in \mathcal{P}_3(X_i)$, the value $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\mathsf{min}}(G_i, \mathcal{X}_i)$.

Let $\mathbf{G}'$ and $\mathbf{X}'$ be the boundaried graphs obtained from $\mathbf{G}$ and $\mathbf{X}$ respectively, by adding two new isolated vertices u_1 and u_2 in the boundary (with unused labels). Let $\mathcal{A}' = (S, X^1 \cup \{u_1\}, X^2 \cup \{u_2\})$. This operation is done to ensure that $X_i' = X_i \cup \{u_i\}$ is non-empty for $i \in [2]$. Obviously, $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G}' \oplus \mathbf{F}, \mathcal{A}')$.

Let $v_1, \ldots, v_{|B|}$ be the vertices of B. For $i \in [|B|]$, let $I_i = \{j \in [d] \mid X_j \setminus A = \{v_i\}\}$. Let $I_0 = \{j \in [d] \mid X_j \subseteq A\}$. Obviously, $(I_i)_{i\in[0,|B|]}$ is a partition of $[d]$.

Let $(\mathbf{H}_{-1}, S_{-1}, s_{-1}, E_{-1}) = (\mathbf{G}', S, 0, \emptyset)$. For i going from 0 up to $|B|$, we construct $(\mathbf{H}_i, S_{i-1}, s_i, E_i)$ from $(\mathbf{H}_{i-1}, S_i, s_{i-1}, E_{i-1})$ such that for any boundaried graph $\mathbf{F}$ with underlying graph F and compatible with $\mathbf{G}$,

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_i \oplus \mathbf{F}_i, \mathcal{A}_i) + s_i,$$

where $\mathbf{F}_i$ is the boundaried graph obtained from $\mathbf{F}$ by adding the edges in E_i and $\mathcal{A}_i = (S_i, X^1 \cup \{u_1\}, X^2 \cup \{u_2\})$. This is obviously true for $i = -1$.

Let $i \in [0, |B|]$. By induction, we have

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}_{i-1}, \mathcal{A}_{i-1}) + s_{i-1}.$$

Let $\mathbf{G}_i' = \boxplus_{j\in I_i} \mathbf{G}_j$. Let $\mathbf{H}_i'$ be the boundaried graph with underlying graph H_i' such that $\mathbf{H}_{i-1} = \mathbf{H}_i' \boxplus \mathbf{G}_i'$.

Suppose first that $i = 0$. Let $\mathcal{P}_0 = (S^0, X_1^0, X_2^0) \in \mathcal{P}_3(X \cup \{u_1, u_2\})$ be such that $\mathcal{A}' \subseteq \mathcal{P}_0$ and $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G}' \oplus \mathbf{F}, \mathcal{A}') = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G}' \oplus \mathbf{F}, \mathcal{P}_0)$. According to Lemma 9.5.14,

$$\begin{aligned}
\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G}' \oplus \mathbf{F}, \mathcal{P}_0) &= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{P}_0) + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G', \mathcal{P}_0) - |S^0 \cap X| \\
&= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(F, \mathcal{P}_0) - |S^0 \cap X| + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(H_0', \mathcal{P}_0) \\
&\quad + \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G_j, \mathcal{P}_0 \cap X_j) - |S^0 \cap X_j|) \\
&= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_0' \oplus \mathbf{F}, \mathcal{P}_0) + \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G_j, \mathcal{P}_0 \cap X_j) - |S \cap X_j|), \\
&\qquad \text{because } S^0 \setminus S \subseteq B \text{ and } X_j \cap B = \emptyset.
\end{aligned}$$

Since this is the case for all such $\mathcal{P}_0$, it implies that

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G}' \oplus \mathbf{F}, \mathcal{A}') = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_0' \oplus \mathbf{F}, \mathcal{A}') + \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G_j, \mathcal{P}_0 \cap X_j) - |S \cap X_j|).$$

Therefore, if $\mathbf{H}_0 = \mathbf{H}_0'$, $S_0 = S$, $s_0 = \sum_{j \in I_0} (\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G_j, \mathcal{P}_0 \cap X_j) - |S \cap X_j|)$, and $E_0 = \emptyset$, then

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_0 \oplus \mathbf{F}_0, \mathcal{A}_0) + s_0.$$

Otherwise, $i \in [|B|]$ and $X_j \setminus A = \{v_i\}$ for each $j \in I_i$. Let $X_i' = \bigcup_{j \in I_i} X_j$. Let $\mathcal{X}_i^S = (S_{i-1} \cup \{v_i\}, X^1 \cup \{u_1\}, X^2 \cup \{u_2\})$, $\mathcal{X}_i^1 = (S_{i-1}, X^1 \cup \{u_1, v_i\}, X^2 \cup \{u_2\})$, and $\mathcal{X}_i^2 = (S_{i-1}, X^1 \cup \{u_1\}, X^2 \cup \{u_2, v_i\})$. For $a \in \{S, 1, 2\}$, let $s_i^a = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G_i', \mathcal{X}_i^a \cap X_i')$. By Lemma 9.5.14,

$$s_i^S = \sum_{j \in I_i} (\hat{p}_{f_{\mathsf{oct}},\min}(G_j, \mathcal{X}_i^S \cap X_j) - |S_{i-1} \cap X_j| - 1) + |S_{i-1} \cap X_i'| + 1$$

and, for $a \in \{1, 2\}$,

$$s_i^a = \sum_{j \in I_i} (\hat{p}_{f_{\mathsf{oct}},\min}(G_j, \mathcal{X}_i^a \cap X_j) - |S_{i-1} \cap X_j|) + |S_{i-1} \cap X_i'|.$$

Therefore, s_i^a can be computed for $a \in \{S, 1, 2\}$ in time $\mathcal{O}(|A| \cdot |I_i|)$. For $a \in \{1, 2\}$, let $\mathbf{H}_i^a$ and $\mathbf{F}_i^a$ be the boundaried graphs obtained from $\mathbf{H}_i'$ and $\mathbf{F}_{i-1}$, respectively, by adding the edge $v_i u_a$. Let $\mathcal{P}_i = (S^i, X_1^i, X_2^i) \in \mathcal{P}_3(X \cup \{u_1, u_2\} \setminus \{v_i\})$ be such that $\mathcal{A}_{i-1} \subseteq \mathcal{P}_i$ and $\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}_{i-1}, \mathcal{A}_{i-1}) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}_{i-1}, \mathcal{P}_i)$. Let $\mathcal{P}_i^S = (S^i \cup \{v_i\}, X_1^i, X_2^i)$, $\mathcal{P}_i^1 = (S^i, X_1^i \cup \{v_i\}, X_2^i)$, and $\mathcal{P}_i^2 = (S^i, X_1^i, X_2^i \cup \{v_i\})$. Note that for $a \in \{S, 1, 2\}$, we have $s_i^a = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(G_i', \mathcal{P}_i^a \cap X_i')$. Then, using Lemma 9.5.15,

$$\begin{aligned}
\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}_{i-1}, \mathcal{P}_i) &= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}((\mathbf{H}_i' \boxplus \mathbf{F}_{i-1}) \oplus \mathbf{G}_i', \mathcal{P}_i) \\
&= \begin{cases}
s_S + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_i' \oplus \mathbf{F}_{i-1}, \mathcal{P}_i^S) - \bar{s} - 1 & \text{if } s_i^S \leq s_i^1, s_i^2, \\
s_1 + \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_i' \oplus \mathbf{F}_{i-1}, \mathcal{P}_i) - \bar{s} & \text{if } s_i^1 = s_i^2 < s_i^S, \\
s_1 + \hat{p}_{f_{\mathsf{oct}},\min}(\mathbf{H}_i^2 \oplus \mathbf{F}_i^2, \mathcal{P}_i) - \bar{s} & \text{if } s_i^1 < s_i^S, s_i^2, \text{and} \\
s_2 + \hat{p}_{f_{\mathsf{oct}},\min}(\mathbf{H}_i^1 \oplus \mathbf{F}_i^1, \mathcal{P}_i) - \bar{s} & \text{otherwise.}
\end{cases}
\end{aligned}$$

Since this is the case for any such $\mathcal{P}_i$, we have that

$$\begin{aligned}
\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) &= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_{i-1} \oplus \mathbf{F}_{i-1}, \mathcal{A}_{i-1}) + s_{i-1} \\
&= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_i \oplus \mathbf{F}_i, \mathcal{A}_i) + s_i,
\end{aligned}$$

where

$$(\mathbf{H}_i, S_i, s_i, E_i) = \begin{cases} (\mathbf{H}'_i, S_{i-1} \cup \{v\}, s_{i-1} + s_S - \bar{s} - 1, E_{i-1}) & \text{if } s_i^S \leq s_i^1, s_i^2, \\ (\mathbf{H}'_i, S_{i-1}, s_{i-1} + s_1 - \bar{s}, E_{i-1}) & \text{if } s_i^1 = s_i^2 < s_i^S, \\ (\mathbf{H}_i^2, S_{i-1}, s_{i-1} + s_1 - \bar{s}, E_{i-1} \cup \{v_i u_2\}) & \text{if } s_i^1 < s_i^S, s_i^2, \\ (\mathbf{H}_i^1, S_{i-1}, s_{i-1} + s_2 - \bar{s}, E_{i-1} \cup \{v_i u_1\}) & \text{otherwise.} \end{cases}$$

Let $S_B = S_{|B|} \setminus S \subseteq B$. Let $\mathbf{H}_B = \mathbf{H}_{|B|} - S_B$. We have $(\mathbf{H}_{|B|} \oplus \mathbf{F}_{|B|}) - S_B = (\mathbf{H}_{|B|} \oplus \mathbf{F}) - S_B = (\mathbf{H}_{|B|} - S_B) \triangleright \mathbf{F} = \mathbf{H}_B \triangleright \mathbf{F}$. Observe that

$$\begin{aligned} \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_{|B|} \oplus \mathbf{F}_{|B|}, \mathcal{A}_{|B|}) &= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}((\mathbf{H}_{|B|} \oplus \mathbf{F}_{|B|}) - S_B, \mathcal{A}_{|B|} \setminus S_B) + |S_B| \\ &= \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_B \triangleright \mathbf{F}, \mathcal{A}') + |S_B|. \end{aligned}$$

Hence,

$$\hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{G} \oplus \mathbf{F}, \mathcal{A}) = \hat{\mathsf{p}}_{f_{\mathsf{oct}},\min}(\mathbf{H}_B \triangleright \mathbf{F}, \mathcal{A}') + |S_B| + s_{|B|}.$$

Observe that $\mathbf{H}_B$ is isomorphic to the boundaried graph obtained from $\mathbf{X}$ by adding two new vertices and the at most $|B|$ edges in $E_{|B|}$, and removing the vertices in S_B. Hence, $|V(H_B)| \leq |X|+2$ and $|E(H_B)| \leq |E(G[X])| + |B|$. Moreover, $|\cup\mathcal{A}'| = |\cup\mathcal{A}| + 2$. Suppose that $F - A \in \mathcal{H}$. Observe that, since the edges in $E_{|B|}$ all have one endpoint in $\{u_1, u_2\}$, $(\mathbf{H}_B \triangleright \mathbf{F}) - (\cup\mathcal{A}) - \{u_1, u_2\}$ is isomorphic to $F - A - S_B$. Since $\mathcal{H}$ is hereditary, $(\mathbf{H}_B \triangleright \mathbf{F}) - (\cup\mathcal{A}') \in \mathcal{H}$. Thus, $(\mathbf{H}_B, \mathcal{A}', |S_B| + s_{|B|})$ is an $\mathcal{H}$-nice reduction of $(G, \mathcal{A})$ with respect to Annotated Odd Cycle Transversal.

At each step i, (H_i, S_i, s_i, E_i) is computable in time $\mathcal{O}(|A| \cdot |I_i|)$, so the computation takes time $\mathcal{O}(|A| \cdot d)$. Hence, Annotated Odd Cycle Transversal is $\mathcal{H}$-nice. ☐

In the next lemma, we adapt the seminal proof of Reed, Smith and Vetta [254] that uses iterative compression to solve Annotated Odd Cycle Transversal in FPT-time parameterized by oct.

Given a graph G and two sets $A, B \subseteq V(G)$, an (A, B)-*cut* is a set $X \subseteq V(G)$ such that there are no paths from a vertex in A to a vertex in B in $V(G) \setminus X$.

Lemma 9.5.17. *There is an algorithm that, given a graph G, (a weight function $w : V(G) \to \mathbb{N}$,) and three disjoint sets $S, A, B \subseteq V(G)$, such that $G - (S \cup A \cup B)$ is bipartite, solves* Annotated Odd Cycle Transversal *(resp.* Weighted Annotated Odd Cycle Transversal*) on (G, S, A, B) in time $\mathcal{O}((n + k) \cdot (m + k^2))$, where $k = |A \cup B|$.*

Proof. We can assume that $S = \emptyset$, given that $S^\star$ is an optimal solution for (G, S, A, B) if and only if $S^\star \setminus S$ is an optimal solution for $(G - S, \emptyset, A, B)$.

Let G^+ be the graph obtained from G by joining each $a \in A$ with each $b \in B$. Let $X = A \cup B$. Let (S_1, S_2) be a partition witnessing the bipartiteness of $G - X = G^+ - X$. We construct an auxiliary bipartite graph G' from G^+ as follows. The vertex set of the auxiliary graph is $V' = V(G) \setminus X \cup \{x_1, x_2 \mid x \in X\}$. We maintain a one-to-one correspondence between the edges of G^+ and the edges of G' by the following scheme:

- for each edge e of $G^+ - X$, there is a corresponding edge in G' with the same endpoints,
- for each edge $e \in E(G^+)$ joining a vertex $y \in S_i$ to a vertex $x \in X$, the corresponding edge in G' joins y to x_{3-i}, and
- for each edge $e \in E(G^+)$ joining two vertices $a \in A$ and $b \in B$, the corresponding edge of G' joins a_1 to b_2.

For $i \in \{1,2\}$, let $X_i = \{x_i \mid x \in X\}$, $A_i = \{a_i \mid a \in A\}$, $B_i = \{b_i \mid b \in B\}$. Note that G' is a bipartite graph, witnessed by the partition $(S_1 \cup X_1, S_2 \cup X_2)$. Let $Y_1 = A_1 \cup B_2$ and $Y_2 = A_2 \cup B_1$. Note also that there is no edge joining Y_1 and Y_2, so there exists a (Y_1, Y_2)-cut in G' that is actually contained in $V(G^+) \setminus X$, and hence $V(G) \setminus X$. Let us show that $S^\star \subseteq V(G) \setminus X$ is a (Y_1, Y_2)-cut in G' if and only if $S^\star$ is an odd cycle transversal of G with A on one side of the bipartition and B on the other one.

Claim 9.5.18. *If $S^\star \subseteq V(G) \setminus X$ is a cutset separating Y_1 from Y_2 in G', then $S^\star$ is an odd cycle transversal of G with A on one side of the bipartition and B on the other one.*

Proof of claim. Let C be an odd cycle of G^+. Suppose towards a contradiction that $C \cap S^\star = \emptyset$. $G^+ - X$ is bipartite, so C intersects X. Moreover, we assumed $G^+[X]$ to be bipartite since A and B are two disjoint independent sets, so C intersects $V(G^+) \setminus X$. Hence, if we divide C into paths whose endpoints are in X and whose internal vertices are in $V(G^+) \setminus X$, each such path in G' has either both endpoints in Y_1 or both endpoints in Y_2, since it is not intersected by the cutset $S^\star$. More specifically, each such path in G' has either both endpoints in A_i, or both endpoints in B_i, or one in A_i and one in B_{3-i}, for $i \in \{1,2\}$. A path with both endpoints in A_i or B_i is even, since the internal vertices are alternatively vertices of S_i and S_{3-i}, with the first and the last in S_{3-i}. A path with one endpoint in A_i and one in B_{3-i} is odd by a similar reasoning, but the number of such paths must be even in order to have a cycle. Therefore, the cycle C is even. Hence the contradiction. So $S^\star$ is an odd cycle transversal of G^+. Given that we added in G^+ all edges between A and B and that $A, B \subseteq V(G^+) \setminus S^\star$, A and B belong to different sides of the bipartition. An odd cycle of G is also an odd cycle of G^+, so $S^\star$ is also an odd cycle transversal of G, with A and B on different sides of the bipartition. $\diamond$

Claim 9.5.19. *If $S^\star \subseteq V(G) \setminus X$ is an odd cycle transversal of G with A and B on different sides of the bipartition, then $S^\star$ is a cutset separating Y_1 from Y_2 in G'.*

Proof of claim. Suppose towards a contradiction that there is a path P between Y_1 and Y_2 that does not intersect $S^\star$. Choose P of minimum length. Hence, the internal vertices of P belong to $G - X$. The endpoints u and v of P are such that, either $u \in A_i$ and $v \in A_{3-i}$, or $u \in B_i$ and $v \in B_{3-i}$, or $u \in A_i$ and $v \in B_i$, or $u \in B_i$ and $v \in A_i$, for $i \in \{1,2\}$. By symmetry, we can assume without loss of generality that $u \in A_1$ and $v \in A_2$ or $v \in B_1$. If $v \in A_2$, then P is an odd path since $G - X$ is bipartite. However, since $G - S^\star$ is bipartite, with A on one side of the bipartition, P is an even path. If $v \in B_1$, then P is an even path since $G - X$ is bipartite. However, since $G - S^\star$ is bipartite, with A and B on different sides of the bipartition, P is an odd path. Hence the contradiction. $\diamond$

The graph G' has $n' = n + |X|$ vertices and at most $m' = m + |X|^2/4$ edges. Finding a vertex-cut $S^\star \subseteq V(G) \setminus X$ separating Y_1 form Y_2 in G' of minimum weight can be reduced to the problem of finding a minimum (weighted) edge-cut. To do so, we transform G' into an arc-weighted directed graph G'', by first replacing every edge by two parallel arcs in opposite directions, and then replacing every vertex v of $G' - Y_1 - Y_2$ by an arc $(v_{\mathsf{in}}, v_{\mathsf{out}})$, such that the arcs incoming (resp. outgoing) to v are now incoming to v_{in} (resp. outgoing of v_{out}). We give weight $w(v)$ to $(v_{\mathsf{in}}, v_{\mathsf{out}})$, and weight $w(V(G))+1$ to the other arcs. Then, computing a minimum (weighted) vertex-cut in G' is equivalent to computing a minimum (weighted) edge-cut in G'', which can be done in time $\mathcal{O}(n' \cdot m')$ [198, 246]. Hence, the running time of the algorithm is $\mathcal{O}((n + |X|) \cdot (m + |X|^2))$. $\square$

We apply Lemma 9.5.16 and Lemma 9.5.17 to the dynamic programming algorithm of Theorem 9.4.1 to obtain the following result.

Corollary 9.5.20. *Given a graph G and a bipartite tree decomposition of G of width k, there is an algorithm that solves* Odd Cycle Transversal *on G in time $\mathcal{O}(3^k \cdot k \cdot n \cdot (m + k^2))$.*

9.5.4 Maximum Weighted Cut

Our dynamic programming scheme also works for problems other than graph modification problems such as the Maximum Weighted Cut problem which is defined as follows. For this part, we only give clues about the proof and we refer the reader to [171] for the complete proof.

Maximum Weighted Cut

Input: A graph G and a weight function $w : E(G) \to \mathbb{N}$.
Task: Find an edge cut of maximum weight.

Let H be a graph. We define f_{cut} as the 2-partition-evaluation function where, for every graph G with edge weight w and for every $\mathcal{P} = (X_1, X_2) \in \mathcal{P}_2(V(G))$,

$$f_{\mathsf{cut}}(G, \mathcal{P}) = w(\mathcal{P}) = w(E(X_1, X_2)).$$

Hence, Maximum Weighted Cut is the problem of computing $\mathsf{p}_{f_{\mathsf{cut}},\max}(G)$. We call its annotated extension Annotated Maximum Weighted Cut. In other words, Annotated Maximum Weighted Cut is defined as follows.

Annotated Maximum Weighted Cut

Input: A graph G, a weight function $w : E(G) \to \mathbb{N}$, and two disjoint sets $X_1, X_2 \subseteq V(G)$.
Task: Find an edge cut of maximum weight such that the vertices in X_1 belongs to one side of the cut, and the vertices in X_2 belong to the other side.

We can prove that Annotated Maximum Weighted Cut has the gluing property.

Lemma 9.5.21 (Gluing property)**.** Annotated Maximum Weighted Cut *has the gluing property. More precisely, given two boundaried graphs $\mathbf{F} = (F, B_F, \rho_F)$ and $\mathbf{G} = (G, B_G, \rho_G)$, a weight function $w : E(\mathbf{F} \oplus \mathbf{G}) \to \mathbb{N}$, a set $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ such that $B_F \cap B_G \subseteq X$, and $\mathcal{X} = (X_1, X_2) \in \mathcal{P}_2(X)$, if we set $\bar{w} = w(\mathcal{X} \cap B_F \cap B_G)$, then we have*

$$\hat{\mathsf{p}}_{f_{\mathsf{cut}},\max}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = \hat{\mathsf{p}}_{f_{\mathsf{cut}},\max}(F, \mathcal{X} \cap V(F), w) + \hat{\mathsf{p}}_{f_{\mathsf{cut}},\max}(G, \mathcal{X} \cap V(G), w) - \bar{w}.$$

We can also reduce a graph $\mathbf{F} \oplus \mathbf{G}$ to a graph F' when the boundary of $\mathbf{F}$ and $\mathbf{G}$ has a single vertex v that is not annotated. See Figure 9.12 for an illustration.

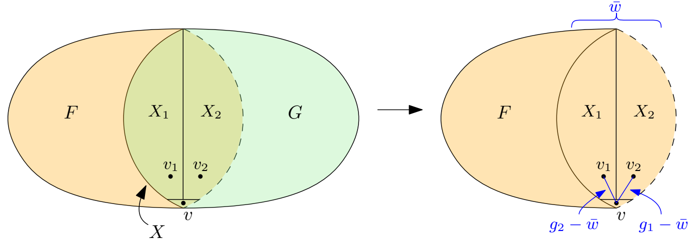

Figure 9.12: Illustration of the gadgetization for Weighted Max Cut.

Lemma 9.5.22 (Gadgetization)**.** *Let $\mathbf{F} = (F, B_F, \rho_F)$ and $\mathbf{G} = (G, B_G, \rho_G)$ be two boundaried graphs. Let $w : E(\mathbf{F} \oplus \mathbf{G}) \to \mathbb{N}$ be a weight function and let $X \subseteq V(\mathbf{F} \oplus \mathbf{G})$ be such that $B_F \cap B_G \subseteq X$. Let also $v \in B_F \cap B_G$ and let $\mathcal{X} = (X_1, X_2) \in \mathcal{P}_2(X \setminus \{v\})$. Suppose that there is $v_1 \in X_1$ and $v_2 \in X_2$ adjacent to v with $w(vv_1) = w(vv_2) = 0$. We set $\mathcal{X}^1 = (X_1 \cup \{v\}, X_2)$ and $\mathcal{X}^2 = (X_1, X_2 \cup \{v\})$. We also set, for $a \in [2]$, $g_a = \hat{\mathsf{p}}_{f_{\mathsf{cut}},\max}(G, \mathcal{X}^a \cap V(G), w)$, we set $\bar{w} = w(\mathcal{X} \cap B_F \cap B_G)$, and we set $w' : E(F) \to \mathbb{N}$ such that $w'(vv_1) = g_2 - \bar{w}$, $w'(vv_2) = g_1 - \bar{w}$, and $w'(e) = w(e)$ otherwise. Then*

$$\hat{\mathsf{p}}_{f_{\mathsf{cut}},\max}(\mathbf{F} \oplus \mathbf{G}, \mathcal{X}, w) = \hat{\mathsf{p}}_{f_{\mathsf{cut}},\max}(F, \mathcal{X}, w').$$

Using Lemma 9.5.21 and Lemma 9.5.22, we can deduce that Annotated Maximum Weighted Cut is $\mathcal{H}$-nice. Essentially, given an instance $(\mathbf{G} = \mathbf{X} \boxplus (\boxplus_{i\in[d]} \mathbf{G}_i), (A, B), \mathcal{A}, w)$, we reduce $\mathbf{G}$ to $\mathbf{X}$ where we add two new vertices in $\mathcal{A}$ and add all edges between these new vertices and the vertices in B. We then show that if the appropriate weight is given to each new edge, then the resulting boundaried graph is an equivalent instance to $\mathbf{G}$ modulo some constant s.

Lemma 9.5.23 (Nice problem)**.** *Let $\mathcal{H}$ be a graph class.* Annotated Maximum Weighted Cut *is $\mathcal{H}$-nice.*

Maximum Weighted Cut is an NP-hard problem [181]. However, there exists a polynomial-time algorithm when restricted to some graph classes. In particular, Grötschel and Pulleyblank [156] proved that Maximum Weighted Cut is solvable in polynomial-time on weakly bipartite graphs, and Guenin [158] proved that weakly bipartite graphs are exactly K_5-odd-minor-free graphs, which gives the following result.

Proposition 9.5.24 ([156, 158])**.** *There is a constant $c \in \mathbb{N}$ and an algorithm that solves* Maximum Weighted Cut *on K_5-odd-minor-free graphs in time $\mathcal{O}(n^c)$.*

Moreover, we observe the following.

Lemma 9.5.25. *A graph G such that $\mathsf{oct}(G) \leq 2$ does not contain K_5 as an odd-minor.*

Proof. Suppose that G contains K_5 as an odd-minor and let η be an odd K_5-expansion of G. Let $u, v \in V(G)$ be such that $G' = G \setminus \{u, v\}$ is bipartite. Given that η has at least three nodes that do not intersect $\{u, v\}$, it implies that K_3 is an odd-minor of G', contradicting its bipartiteness. □

Combining Proposition 9.5.24 and Lemma 9.5.25, we have that Annotated Maximum Weighted Cut is FPT parameterized by oct.

Lemma 9.5.26. *There is an algorithm that, given a graph G, a weight function $w : E(G) \to \mathbb{N}$, and two disjoint sets $X_1, X_2 \subseteq V(G)$, such that $G' = G \setminus (X_1 \cup X_2)$ is bipartite, solves* Annotated Maximum Weighted Cut *on (G, X_1, X_2, w) in time $\mathcal{O}(k \cdot n' + n'^c)$, where $k = |X_1 \cup X_2|$, $n' = |V(G')|$, and c is the constant of Proposition 9.5.24.*

Proof. Let G'' be the graph obtained from G by identifying all vertices in X_1 (resp. X_2) to a new vertex x_1 (resp. x_2). Let $w' : V(G'') \to \mathbb{N}$ be such that $w'(x_1x_2) = \sum_{e\in E(G)} w(e) + 1$, $w'(x_iu) = \sum_{x\in X_i} w(xu)$ for $i \in [2]$ and $u \in N_G(X_i)$, and $w'(e) = w(e)$ otherwise. Let $(X_1^\star, X_2^\star) \in \mathcal{P}_2(V(G))$ be such that $(X_1, X_2) \subseteq (X_1^\star, X_2^\star)$. For $i \in [2]$, let $X_i' = X_i^\star \setminus X_i$. Then

$$\begin{aligned} w(X_1^\star, X_2^\star) &= w(X_1, X_2) + w(X_1', X_2') + \sum_{xy\in E(X_1, X_2')} w(xy) + \sum_{xy\in E(X_1', X_2)} w(xy) \\ &= w(X_1, X_2) + w'(X_1', X_2') + \sum_{u\in X_2\cap N_G(X_1)} w'(x_1u) + \sum_{u\in X_1\cap N_G(X_2)} w'(x_2u) \\ &= w'(X_1' \cup \{x_1\}, X_2' \cup \{x_2\}) + w(X_1, X_2) - w'(x_1x_2) \end{aligned}$$

Let $\bar{w}$ be the constant $w(X_1, X_2) - w'(x_1x_2)$. Hence,

$$f_{\mathsf{cut}}(G, (X_1^\star, X_2^\star)) = f_{\mathsf{cut}}(G'', (X_1' \cup \{x_1\}, X_2' \cup \{x_2\})) + \bar{w},$$

and so $\hat{p}_{f_{\mathsf{cut}},\max}(G, (X_1, X_2)) = \hat{p}_{f_{\mathsf{cut}},\max}(G'', (\{x_1\}, \{x_2\})) + \bar{w}$. Moreover, given that the weight of the edge x_1x_2 is larger than the sum of all other weights, x_1 and x_2 are never on the same side of a maximum cut in G''. Hence, $\hat{p}_{f_{\mathsf{cut}},\max}(G'', (\{x_1\}, \{x_2\})) = \mathsf{p}_{f_{\mathsf{cut}},\max}(G'')$, and therefore,

$$\hat{p}_{f_{\mathsf{cut}},\max}(G, (X_1, X_2)) = p_{f_{\mathsf{cut}},\max}(G'') + \bar{w}.$$

Constructing G'' takes time $\mathcal{O}(k \cdot n)$ and computing $\bar{w}$ takes time $\mathcal{O}(k^2)$. Since $\mathsf{oct}(G'') = 2$, according to Proposition 9.5.24 and Lemma 9.5.25, an optimal solution to Maximum Weighted Cut on G'' can be found in time $\mathcal{O}(n'^c)$, and thus, an optimal solution to Annotated Maximum Weighted Cut on (G, X_1, X_2) can be found in time $\mathcal{O}(k \cdot (k + n') + n'^c)$. □

We can finally apply Lemma 9.5.23 and Lemma 9.5.26 to the dynamic programming algorithm of Theorem 9.4.1 to obtain the following result.

Theorem 9.5.27. *Given a graph G and a bipartite tree decomposition of G of width k, there is an algorithm that solves* Maximum Weighted Cut *on G in time $\mathcal{O}(2^k \cdot (k \cdot (k + n) + n^c))$, where c is the constant of Proposition 9.5.24.*

9.5.5 Hardness of covering problems

For any graph G, it holds that $\mathsf{btw}(G) \leq \mathsf{oct}(G)$. Thus, for a problem Π to be efficiently solvable on graphs of bounded btw, Π needs to be efficiently solvable on graphs of bounded oct, and first and foremost on bipartite graphs. Unfortunately, many problems are NP-complete on bipartite graphs (or on graphs of small oct), and hence $\mathsf{para\text{-}NP}$-complete parameterized by btw.

Vertex Deletion to $\mathcal{H}$ is known to be NP-complete on general graphs for every non-trivial graph class $\mathcal{H}$ [216]. However, for some graph classes $\mathcal{H}$, it might change when we restricte the input graph to be bipartite. Yannakakis [314] characterizes hereditary graph classes $\mathcal{H}$ for which Vertex Deletion to $\mathcal{H}$ on bipartite graphs is polynomial-time solvable and those for which Vertex Deletion to $\mathcal{H}$ remains NP-complete.

A problem Π is said to be *trivial* on a graph class $\mathcal{H}$ if the solution to Π is the same for every graph $G \in \mathcal{H}$. Otherwise, Π is called *nontrivial* on $\mathcal{H}$. Given a graph G, let $\nu(G) = |\{N_G(v) \mid v \in V(G)\}|$. Given a graph class $\mathcal{H}$, let $\nu(\mathcal{H}) = \sup\{\nu(G) \mid G \in \mathcal{H}\}$.

Proposition 9.5.28 ([314])**.** *Let $\mathcal{H}$ be a hereditary graph class such that* Vertex Deletion to $\mathcal{H}$ *is nontrivial on bipartite graphs.*

- *If $\nu(\mathcal{H}) = +\infty$, then* Vertex Deletion to $\mathcal{H}$ *is* NP*-complete on bipartite graphs.*
- *If $\nu(\mathcal{H}) < +\infty$, then* Vertex Deletion to $\mathcal{H}$ *is polynomial time-solvable on bipartite graphs.*

Hence, here is a non-exhaustive list of problems that are NP-complete on bipartite graphs: Vertex Deletion to $\mathcal{H}$ where $\mathcal{H}$ is a minor-closed graph class that contains edges (and hence Feedback Vertex Set, Vertex Planarization, H-Minor-Cover for H containing P_3 as a subgraph), H-Subgraph-Cover, H-Induced-Subgraph-Cover, and H-Odd-Minor-Cover for H bipartite graph containing P_3 as a (necessarily induced) subgraph, Vertex Deletion to graphs of degree at most p for $p \geq 1$, Vertex Deletion to graphs of girth at least p for $p \geq 6$ (note that the smallest non-trivial lower bound on the length of a cycle in a bipartite graph is six, or equivalently five).

As a consequence of the above results, all the above problems, when parameterized by bipartite treewidth, are $\mathsf{para\text{-}NP}$-complete.

CHAPTER 10

$\mathcal{H}$-planarity and beyond

Contents

In this chapter, we prove the results presented in Section 2.6, that are restated here for convenience.

Theorem 2.6.1. *Let $\mathcal{H}$ be a graph class that is hereditary, CMSO-definable, and decidable in time $\mathcal{O}(n^c)$ for some constant c. Then, there is an algorithm solving $\mathcal{H}$-Planarity in time $\mathcal{O}(n^4 + n^c \log n)$.*

Theorem 2.6.2. *Let $\mathcal{H}$ be a hereditary and CMSO-definable graph class that is closed under disjoint union. Suppose that there is an* FPT*-algorithm solving* Vertex Deletion to $\mathcal{H}$ *parameterized by the solution size k in time $\mathcal{O}_k(n^c)$. Then there is an* FPT*-algorithm that, given a graph G and $k \in \mathbb{N}$, decides whether G has $\mathcal{H}$-planar treedepth at most k in time $\mathcal{O}_k(n^4 + n^c \log n)$.*

Theorem 2.6.3. *Let $\mathcal{H}$ be a hereditary and CMSO-definable graph class that is closed under disjoint union. Suppose that there is an* FPT*-algorithm solving* Vertex Deletion to $\mathcal{H}$ *parameterized by the solution size k in time $\mathcal{O}_k(n^c)$. Then there is an* FPT*-algorithm that, given a graph G and $k \in \mathbb{N}$, decides whether G has $\mathcal{H}$-planar treewidth at most k in time $\mathcal{O}_k(n^4 + n^c \log n)$.*

The outline of the algorithms is presented in Section 10.1, while Theorem 2.6.1, Theorem 2.6.2, and Theorem 2.6.3 are presented in Section 10.2, Section 10.3, and Section 7.6, respectively. Several applications are proposed in Section 10.5. Finally, the necessity of some of the conditions on the target class are discussed in Section 10.6.

Some conventions. In this chapter, we only consider sphere decompositions that are *vortex-free*. Therefore, for convenience, we use the shortcut sphere decompositions and renditions to denote vortex-free sphere decompositions and vortex-free renditions in the sphere, respectively. Also, in this chapter, the class of planar graphs is denoted by $\mathcal{P}$.

Let us begin with some definitions.

Planar treewidth. The *planar treewidth* of a graph G, denoted by $\mathsf{ptw}(G)$, is the minimum k such that there exists a tree decomposition of G such that each bag either has size at most $k+1$ or has a planar torso.

Planar treedepth. Recall from Section 1.5 that the *treedepth* of a graph G, denoted by $\mathsf{td}(G)$, is zero if G is the empty graph, and one plus the minimum treedepth of the graph obtained by removing one vertex from each connected component of G otherwise. The *planar treedepth* of G, denoted by $\mathsf{ptw}(G)$, is defined as the treedepth, but where we remove a planar modulator from each connected component of G instead of a vertex. Recall that the formal definition is given in Section 2.6. The fact that $\mathsf{ptd}(G) \leq k$ is certified by a sequence $X_1, \dots, X_k$ of successive planar modulators that need to be removed. We refer to such a sequence as a *certifying elimination sequence* (see Figure 2.3).

$\mathcal{G} \triangleright \mathcal{H}$-modulators. Let $\mathcal{H}$ and $\mathcal{G}$ be graph classes. We define $\mathcal{G} \triangleright \mathcal{H}$ to be the class of graphs G who contains a vertex subset $X \subseteq V(G)$, called $\mathcal{G} \triangleright \mathcal{H}$-*modulator*, such that $\mathsf{torso}(G, X) \in \mathcal{G}$ and, for each $C \in \mathsf{cc}(G - X)$, $C \in \mathcal{H}$. Thus, $\mathcal{P} \triangleright \mathcal{H}$ is the class of $\mathcal{H}$-planar graphs. Note that the operator $\triangleright$ is associative, i.e. $\mathcal{F} \triangleright (\mathcal{G} \triangleright \mathcal{H}) = (\mathcal{F} \triangleright \mathcal{G}) \triangleright \mathcal{H}$. For $k \in \mathbb{N}_{\geq 1}$, we set $\mathcal{G}^{k+1} = \mathcal{G} \triangleright \mathcal{G}^k$, where $\mathcal{G}^1 = \mathcal{G}$. In particular, $\mathcal{P}^k$ is the class of graphs of planar treedepth at most k and $\mathcal{P}^k \triangleright \mathcal{H}$ is the class of graphs with $\mathcal{H}$-planar treedepth at most k.

Note that, given $k \in \mathbb{N}$, if $\mathcal{G}_k$ is class of graphs with treewidth (resp. treedepth / size / planar treewidth / planar treedepth) at most k, then $\mathcal{G}_k \triangleright \mathcal{H}$ is the class of graphs with $\mathcal{H}$-treewidth (resp. elimination distance to $\mathcal{H}$ / $\mathcal{H}$-deletion[1] / $\mathcal{H}$-planar treewidth / $\mathcal{H}$-planar treedepth) at most k. The class of graphs with planar treewidth at most k will be denoted by $\mathcal{PT}_k$.

[1]If $\mathcal{H}$ is closed under disjoint union.

Counting Monadic Second-Order Logic. Recall that CMSO logic is defined in Section 1.6. Planarity and connectivity are expressible in CMSO logic, see e.g. [68, Subsection 1.3.1]. Additionaly, $\mathsf{torso}(G, X)$ is the graph with vertex set X, and edge set the pairs $(u, v) \in X^2$ such that either $uv \in E(G)$ or u and v are connected in $G - (X \setminus \{u, v\})$, which is also easily expressible in CMSO logic. Therefore, we observe the following.

Observation 10.0.1. *If $\mathcal{H}$ is a CMSO-definable graph class, then $\mathcal{H}$-Planarity is expressible in CMSO logic.*

10.1 The algorithms

In Subsection 10.1.1, we prove Theorem 2.6.1, Theorem 2.6.2, and Theorem 2.6.3, assuming some results that will be proved later in the chapter but that we sketch in Subsection 10.1.2 (for $\mathcal{H}$-planar and $\mathcal{H}$-planar treewidth) and Subsection 10.1.3 (for $\mathcal{H}$-planar treedepth).

10.1.1 The algorithms

The starting point of our algorithmms is the result of Lokshtanov, Ramanujan, Saurabh, and Zehavi [224] reducing a CMSO-definable graph problem to the same problem on unbreakable graphs.

Unbreakable graphs. Let G be a graph and let $c, s \in \mathbb{N}$. If there exists a separation (X, Y) of order at most c such that $|X \setminus Y| \geq s$ and $|Y \setminus X| \geq s$, called an *(s, c)-witnessing separation*, then G is *(s, c)-breakable*. Otherwise, G is *(s, c)-unbreakable*.

Proposition 10.1.1 (Theorem 1, [224])**.** *Let ψ be a CMSO sentence and let $d > 4$ be a positive integer. There exists a function $\alpha : \mathbb{N} \to \mathbb{N}$, such that, for every $c \in \mathbb{N}$, if there exists an algorithm that solves $\mathrm{CMSO}[\psi]$ on $(\alpha(c), c)$-unbreakable graphs in time $\mathcal{O}(n^d)$, then there exists an algorithm that solves $\mathrm{CMSO}[\psi]$ on general graphs in time $\mathcal{O}(n^d)$.*

In our case, we have the following observation.

Observation 10.1.2. *Let $\mathcal{H}$ and $\mathcal{G}$ be graph classes. If every graph in $\mathcal{G}$ is K_{k+1}-minor-free, then, for any $\mathcal{G} \triangleright \mathcal{H}$-modulator X of a graph G and for any $C \in \mathsf{cc}(G - X)$, $|N_G(V(C))| \leq k$. In particular, if $\mathcal{G} = \mathcal{P}$, then $|N_G(V(C))| \leq 4$, if $\mathcal{G} = \mathcal{P}^k$, then $|N_G(V(C))| \leq 4k$, and if $\mathcal{G} = \mathcal{PT}_k$, then $|N_G(V(C))| \leq \max\{k + 1, 4\}$.*

Let $a, k \in \mathbb{N}$, $\mathcal{H}$ and $\mathcal{G}$ be two graph classes such that every graph in $\mathcal{G}$ is K_{k+1}-minor-free, and G be an (a, k)-unbreakable graph. We say that a $\mathcal{G} \triangleright \mathcal{H}$-modulator X of a graph G is a *big-leaf $\mathcal{G} \triangleright \mathcal{H}$-modulator* of G is there is exists a (unique) component $D \in \mathsf{cc}(G - X)$ of size at least a, called *big leaf with respect to X*.

Therefore, if we work on $(\alpha(4), 4)$-unbreakable graphs, then given a planar $\mathcal{H}$-modulator S of a graph G, either S is a big-leaf planar $\mathcal{H}$-modulator of G, that is, there is a unique component $C \in \mathsf{cc}(G - S)$ such that $|V(C)| \geq \alpha(4)$ and $|V(G) \setminus V(C)| < \alpha(4) + |N_G(V(C))|$ or for each $C \in \mathsf{cc}(G - S)$, $|V(C)| < \alpha(4)$. In this chapter, we will thus solve $\mathcal{H}$-Planarity on $(\alpha(4), 4)$-unbreakable graphs, which, by applying Proposition 10.1.1, immediately implies Theorem 2.6.1.

More specifically, we will split $\mathcal{H}$-Planarity into two complementary subproblems. The first one is Big-Leaf $\mathcal{H}$-Planarity, which is defined as follows.

Big-Leaf $\mathcal{H}$-Planarity

Input: A graph G.
Question: Does G admit a planar $\mathcal{H}$-modulator S such that there is $D \in \mathsf{cc}(G-S)$ of size at least $\alpha(4)$?

This problem is easy to solve using a brute-force method.

Lemma 10.1.3. *Let $\mathcal{H}$ be a polynomial-time decidable graph class. Then there is an algorithm that solves* Big-Leaf $\mathcal{H}$-Planarity *on $(\alpha(4), 4)$-unbreakable graphs in polynomial time.*

Proof. We enumerate in polynomial time all separations (A, B) of G of order at most four such that $G[A \setminus B]$ is connected and $|B \setminus A| < \alpha(4)$. For each such separation (A, B), we consider all sets S with $A \cap B \subseteq S \subseteq B$ (there are at most $2^{\alpha(4)-1}$ such sets). For every S, we check whether the torso of S is planar and that each connected component of $G - S$ belongs to $\mathcal{H}$. If there is such a set S and that one connected component of $G - S$ has size at least $\alpha(4)$, then we conclude that G is a yes-instance of Big-Leaf $\mathcal{H}$-Planarity. If for each such (A, B) and each such S, we did not report a yes-instance, then we report a no-instance. These checks take constant time given that $|B \setminus A| < \alpha(4)$. This concludes the proof. $\square$

The second subproblem is the following.

Small-Leaves $\mathcal{H}$-Planarity

Input: A graph G.
Question: Does G admit a planar $\mathcal{H}$-modulator S such that, for each $D \in \mathsf{cc}(G-S)$, $|V(D)| < \alpha(4)$?

Let $k \in \mathbb{N}$ and $\mathcal{H}$ be a graph class. Recall that $\mathcal{H}^{(k)}$ is the subclass of $\mathcal{H}$ containing graphs with at most k vertices. In this setting, Small-Leaves $\mathcal{H}$-Planarity is exactly $\mathcal{H}^{(\alpha(4)-1)}$-Planarity. More generally than Small-Leaves $\mathcal{H}$-Planarity, we prove that $\mathcal{H}^{(k)}$-Planarity is solvable in FPT-time parameterized by k in the following theorem (see Subsection 10.2.6 for the proof).

Theorem 10.1.4. *Let $k \in \mathbb{N}$ and let $\mathcal{H}$ be a polynomial-time decidable hereditary graph class. Then there is an algorithm that solves $\mathcal{H}^{(k)}$-*Planarity *in time $f(k) \cdot n(n+m)$ for some computable function f.*

Given that a graph G is a yes-instance of $\mathcal{H}$-Planarity if and only if it is a yes-instance of at least one of Big-Leaf $\mathcal{H}$-Planarity and Small-Leaves $\mathcal{H}$-Planarity, Theorem 2.6.1 immediately follows from Proposition 10.1.1, Lemma 10.1.3, and Theorem 10.1.4.

We do exactly the same for $\mathcal{H}$-planar treedepth and $\mathcal{H}$-planar treewidth, but here, by Observation 10.1.2, we consider $(\alpha(k'), k')$-unbreakable graphs for $k' = 4k$ and $k' = \max\{4, k+1\}$, respectively, instead of $k' = 4$.

In this case, similarly to Lemma 10.1.3, we prove the following.

Lemma 10.1.5. *Let $\mathcal{H}$ be a hereditary graph class that is closed under disjoint union and such that there is an* FPT*-algorithm that solves* Vertex Deletion to $\mathcal{H}$ *parameterized by the solution size h in time $f(h) \cdot n^c$. Let $a, k \in \mathbb{N}$. Let $\mathcal{G}_k$ be the class of graphs with planar treewidth (resp. treewidth / planar treedepth / treedepth) at most k. Let $k' := \max\{4, k+1\}$ (resp. $k+1$ / $4k$ / k). Then there is an algorithm that, given an (a, k')-unbreakable graph G, either reports that G has no big-leaf $\mathcal{G}_k \triangleright \mathcal{H}$-modulator, or outputs a $\mathcal{G}_k \triangleright \mathcal{H}$-modulator of G, in time $f(k) \cdot 2^{\mathcal{O}((a+k)^2)} \cdot \log n \cdot (n^c + n + m)$.*

The proof of Lemma 10.1.5 is based on random sampling technique from [60] (Proposition 10.3.1). Assuming G has a big-leaf $\mathcal{G}_k \triangleright \mathcal{H}$-modulator X with big leaf D, we guess a set $U \subseteq V(G)$ such that $N_G(V(D)) \subseteq U$ and $V(G) \setminus V(D) \subseteq V(G) \setminus U$, and deduce X and D from this guess. See Subsection 10.3.1 for the proof.

Meanwhile, similarly to Theorem 10.1.4, we prove the two following results, whose proofs can be found respectively in Subsection 10.3.2 and Subsection 10.4.2.

Lemma 10.1.6. *Let $\mathcal{H}$ be a graph class that is hereditary. Let $a, k \in \mathbb{N}$, $k' = 4k$. Then there is an algorithm that, given an (a, k')-unbreakable graph G, check whether G has $\mathcal{H}^{(a-1)}$-planar treedepth at most k in time $\mathcal{O}_{k,a}(n \cdot (n+m))$.*

Lemma 10.1.7. *Let $\mathcal{H}$ be a graph class that is hereditary and closed under disjoint union. Let $a, k \in \mathbb{N}$. Then there is an algorithm that, given an $(a, 3)$-unbreakable graph G, check whether G has $\mathcal{H}^{(a-1)}$-planar treewidth at most k in time $\mathcal{O}_{k,a}(n \cdot (n+m))$.*

Therefore, Theorem 2.6.2 immediately follows from Proposition 10.1.1, Lemma 10.1.5, and Lemma 10.1.6, and Theorem 2.6.3 immediately follows from Proposition 10.1.1, Lemma 10.1.5, and Lemma 10.1.7.

Remark. For ease of presentation, the algorithm we proposed in Lemma 10.1.3 while simple, does not have an optimal running time. We can actually use instead an algorithm similar to the one of Lemma 10.1.5 that solves Big-Leaf $\mathcal{H}$-Planarity in time $\mathcal{O}((n^c + n + m)\log n)$, where Membership to $\mathcal{H}$ is decidable in time $\mathcal{O}(n^c)$ for some constant c. Thus, we can also solve $\mathcal{H}$-Planarity in time $\mathcal{O}(n^4 + n^c \log n)$.

10.1.2 Outline of our technique of $\mathcal{H}$-planarity (and $\mathcal{H}$-planar treewidth)

The proof of Theorem 10.1.4 for $\mathcal{H}$-planar graphs is quite involved, and in particular, requires the introduction of many notions. In this section, we sketch the proof, and the formal proof with all details is deferred to Section 10.2. Also, as we will argue at the end of this part, the proof of Lemma 10.1.7 for graphs of bounded $\mathcal{H}$-planar treewidth is very similar to the one of Theorem 10.1.4. For the sake of a good understanding, we give here informal definitions of the necessary notions.

The first step of the algorithm is to find a flat wall in G. For this, one would usually use the classical Flat Wall theorem (see Proposition 4.6.3 and also [141, 194, 271, 286]) that says that either G contains a big clique as a minor, or G has bounded treewidth, or there is a set $A \subseteq V(G)$, called *apex set*, of bounded size and a wall W that is flat in $G - A$. However, in our case, we want W to be flat in G with no apex set. Therefore, we use a variant of the Flat Wall theorem that is implicit in [141, Theorem 1] and [205, Lemma 4.7]. However, we are not aware of an algorithmic statement of this variant, and so we prove it here (see Theorem 10.2.2). We thus prove that there is a function f and an algorithm that, given a graph G and $h, r \in \mathbb{N}$, outputs one of the following:

- **Case 1:** a report that G contains an apex $(h \times h)$-grid as a minor, or
- **Case 2:** a tree decomposition of G of width at most $f(h) \cdot r$, or
- **Case 3:** a wall W of height r that is flat in G and whose compass has treewidth at most $f(h) \cdot r$.

We apply the above algorithm to the instance graph G of $\mathcal{H}^{(k)}$-Planarity for $h, r = \Theta(\sqrt{k})$ (precise values are given in Subsection 10.2.6) and consider three cases depending on the output in the **next step**.

Case 1: If G contains the apex grid of height h as a minor, then we are able to argue that G is a no-instance (Lemma 10.2.3). This is easy if the apex grid is a subgraph of G as for any separation of the apex grid of order at most four, there is a nonplanar part whose size is bigger than k, for our choice of h. We use this observation to show that the same holds if the apex grid is a minor.

Case 2: If G has a tree decomposition of bounded treewidth, then we can use Courcelle's theorem to solve the problem. Recall that Courcelle's theorem (Proposition 4.3.2) says that if a problem can be defined in CMSO logic, then it is solvable in FPT-time on graphs of bounded treewidth. In our case, observe that, in Theorem 10.1.4, we do not ask for $\mathcal{H}$ to be CMSO-definable. Nevertheless, given that there is a finite number of graphs of size at most k, $\mathcal{H}^{(k)}$ is trivially CMSO-definable by enumerating all the graphs in $\mathcal{H}^{(k)}$. Then, $\mathcal{H}^{(k)}$-Planarity is easily expressible in CMSO logic (Observation 10.0.1), hence the result.

It remains to consider the third and most complicated case.

Case 3: There is a flat wall W in G of height r whose compass G' has treewidth upper bounded by $f(h) \cdot r$. Then we apply Courcelle's theorem again, on G' this time, since it has bounded treewidth. If G' is not $\mathcal{H}^{(k)}$-planar, then neither is G because $\mathcal{H}$ is hereditary, so we conclude that G is a no-instance of $\mathcal{H}^{(k)}$-Planarity. Otherwise, G' is $\mathcal{H}^{(k)}$-planar. Let v be a central vertex of W. Then we prove that v is an *irrelevant vertex* in G, i.e. a vertex such that G and $G - v$ are equivalent instances of the problem. Therefore, we call our algorithm recursively on the instance $G - v$ and return the obtained answer.

The description of our algorithm in Case 3 is simple. However, proving its correctness is the crucial and most technical part of this chapter which deviates significantly from other applications of the irrelevant vertex technique. The reason for this is that the modulator is a planar graph that might have big treewidth and might be spread "all around the input graph", therefore it does not satisfy any locality condition that might make possible the application of standard irrelevant-vertex arguments (such as those crystalized in algorithmic-meta-theorems in [120, 146, 287]). Most of the technical part of this chapter is devoted to dealing with this situation where the modulator is not anymore "local".

It is easy to see that if G has a planar $\mathcal{H}^{(k)}$-modulator, then the same holds for G' and $G - v$ as $\mathcal{H}$ is hereditary. The difficult part is to show the opposite implication.

First, we would like to emphasize that in the standard irrelevant vertex technique of Robertson and Seymour [271], the existence of a big flat wall W with some specified properties implies that a central vertex is irrelevant. Here, we cannot make such a claim because the compass of W may be nonplanar and we cannot guarantee that v does not belong to $G - X$ for a (potential) planar $\mathcal{H}^{(k)}$-modulator X. Therefore, we have to verify that G' is a yes-instance before claiming that v is irrelevant. Then we have to show that if both G' and $G - v$ have planar $\mathcal{H}^{(k)}$-modulators, then G also has a planar $\mathcal{H}^{(k)}$-modulator. The main idea is to combine modulators for G' and $G - v$ and construct a modulator for G. However, this is nontrivial because the choice of a connected component C which is outside of a modulator restricts the possible choices of other such components and this may propagate arbitrarily around a rendition. In fact, this propagation effect is used in the proof of Theorem 10.6.1 where we show that $\mathcal{H}$-Planarity is NP-complete for nonhereditary classes. Still, for hereditary classes $\mathcal{H}$, we are able to show that if both G' and $G - v$ are yes-instances of $\mathcal{H}^{(k)}$-Planarity, then these instances have some particular solutions that could be glued together to obtain a planar $\mathcal{H}^{(k)}$-modulator for G. In the remaining part of this section, we informally explain the choice of compatible solutions.

$\mathcal{H}$-compatible renditions. Let G be an $\mathcal{H}$-planar graph. Let X be a planar $\mathcal{H}$-modulator in G, and Γ be a planar embedding of the torso of X. For each component D in $G - X$, the neighborhood of D induces a clique of size at most four in the torso of X. Therefore, it is contained in a disk Δ_D in Γ whose boundary is the outer face of the clique, that is a cycle of size at most three. Therefore, for each pair of such disks, either one is included in the other, or their interior is disjoint. If we take all *inclusion-wise maximal* such disks, along with the rest of the embedding disjoint from these disks, this then essentially defines the set of cells $\mathcal{D}$ of a rendition ρ of G. In particular, for each cell c of ρ, the graph induced by c has a planar $\mathcal{H}$-modulator X_c such that the torso of X_c admits a planar embedding with the vertices of $\tilde{c}$ on the outer face. Such a rendition ρ (resp. cell c) is said to be *$\mathcal{H}$-compatible*, and it is direct to check that if G admits an $\mathcal{H}$-compatible rendition, then G is $\mathcal{H}$-planar. This is what we prove, with additional constraints and more formally, in Lemma 10.2.5. In particular, we talk here in terms of renditions for simplicity, while we actually use sphere decompositions, which are similar to renditions, but defined on the sphere instead of a disk, with no cyclic permutation Ω defining a boundary. See Figure 10.1 for an illustration and Subsection 10.2.3 for an accurate definition.

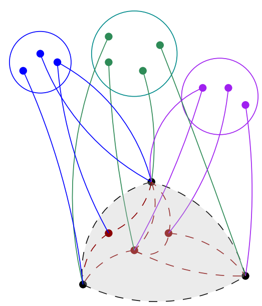

Figure 10.1: An $\mathcal{H}$-compatible cell c (in gray). The $\mathcal{H}$-modulator X is the set of red and black vertices, with the black vertices being the vertices of $\tilde{c}$. The blue, green, and purple balls are the connected components obtained after removing X, and the dashed lines are used for the edges of the torso that are not necessarily edges of the graph.

Therefore, since G' and $G - v$ are $\mathcal{H}^{(k)}$-planar, there are a rendition ρ' of G' and a rendition ρ_v of $G - v$ that are $\mathcal{H}^{(k)}$-compatible. A way to prove that G is $\mathcal{H}^{(k)}$-planar would then be to glue these two renditions together to find an $\mathcal{H}^{(k)}$-compatible rendition ρ^* of G. For instance, we may take a disk Δ in the rendition corresponding to the flat wall W with $v \in \Delta$ and define ρ^* to be the rendition equal to ρ' when restricted to Δ, and equal to ρ_v outside of Δ. A major problem in such an approach is however that it may only work if we are able to guarantee that the cells of ρ' and ρ_v are pairwise disjoint on the boundary of Δ. Given that a graph G may have many distinct renditions, the ideal would be to manage to find a *unique* rendition of G that is minimal or maximal in some sense and could then guarantee that its cells are crossed[2] by no cell of another rendition of G. Unfortunately, we do not have such a result; in fact, we have no way to "repair" these renditions so as to achieve cell-compatibility. Instead, we prove a new property of renditions that allows us to glue renditions together.

Well-linked and ground-maximal renditions. Recall that a rendition ρ is well-linked if every cell c of ρ has $|\tilde{c}|$ vertex disjoint paths to $V(\Omega)$, where $\tilde{c}$ is the set of vertices on the boundary of c. A rendition ρ is *more grounded* than a rendition ρ' if every cell of ρ is contained in a cell of ρ', or

[2] A cell c_1 of a rendition ρ_1 *crosses* a cell c_2 of a rendition ρ_2 if their vertex sets have a non-empty intersection, but that neither of them is contained in the other.

equivalently, if ρ can be obtained from ρ' by splitting cells to ground more vertices. A rendition ρ is *ground-maximal* if no rendition is more grounded than ρ. See Subsection 10.2.3 and Subsection 10.2.4 for the accurate definition of ground-maximality and well-linkedness respectively.

Let (G, Ω) be a society. What we prove is that for any ground-maximal rendition ρ_1 of (G, Ω) and for any well-linked rendition ρ_2 of (G, Ω), ρ_1 is always more grounded than ρ_2, or, in other words, every cell c_1 of ρ_1 is contained in a cell c_2 of ρ_2 (see Corollary 10.2.14). To prove this, we assume towards a contradiction that either c_1 and c_2 cross, or that c_2 is contained in c_1. In both cases, we essentially prove that we can replace c_1 by the cells obtained from the restriction of ρ_2 in c_1, which gives a rendition ρ^* that is more grounded than ρ_1, contradicting the ground-maximality of ρ_1. See Figure 10.2 for some illustration.

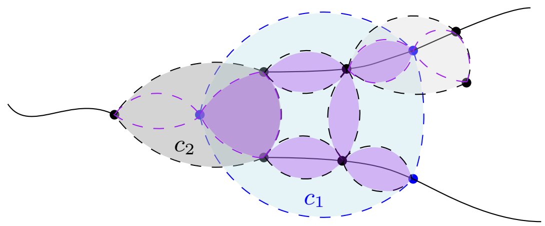

Figure 10.2: c_2 is a cell of a well-linked rendition ρ_2 (in gray). If c_2 crosses a cell c_1 of another (blue) rendition ρ_1, then there is a new (purple) rendition ρ^* that is more grounded than the blue one.

Going back to G' and $G - v$, given that $\mathcal{H}$, and thus $\mathcal{H}^{(k)}$, is *hereditary*, it implies that any rendition that is more grounded than an $\mathcal{H}^{(k)}$-compatible rendition is also $\mathcal{H}^{(k)}$-compatible. Therefore, we can assume ρ' and ρ_v to be ground-maximal. Additionally, by definition, there is a rendition ρ of (G', Ω) witnessing that W is a flat wall of G, where $V(\Omega)$ is a vertex subset of the perimeter of W. Then, by known results (Proposition 10.2.10), we can assume ρ to be well-linked.

Therefore, the cells of ρ' and ρ_v are contained in cells of ρ when restricted to $G' - v$. Actually, it is a bit more complicated given that, while ρ is a rendition of the society (G', Ω), we just know that ρ' and ρ_v are renditions of (G', Ω') and $(G - v, \Omega_v)$ respectively, for some cyclic permutations Ω' and Ω_v that we do not know. What we actually prove, thanks to the structure of the wall, is that the cells of ρ' and ρ_v in $G' - v$ that are far enough from v and the perimeter of W are contained in cells of ρ. For this, rather than Corollary 10.2.14 that says that a ground-maximal rendition of a society is more-grounded than a well-linked rendition of the exactly same rendition, we prefer to use an auxiliary lemma (see Lemma 10.2.13) that says that, given two renditions ρ_1 and ρ_2 of possibly different societies, if a ground-maximal cell c_1 of ρ_1 that does not intersect $V(\Omega)$ intersects only well-linked cells of ρ_2, then c_1 is contained in some cell c_2 of ρ_2.

Given that we found a strip around v in W where the cells of ρ' and ρ_v are contained in cells of ρ, we can pick a disk Δ, whose boundary is in this strip, that intersects ρ only on ground vertices. In other words, the interior of each cell of ρ is either contained in Δ or outside of Δ. But then, the same holds for the cells of ρ' and ρ_v. Therefore, we can finally correctly define the rendition ρ^* of G that is equal to ρ' when restricted to Δ and equal to ρ_v outside of Δ (see Lemma 10.2.15). See Figure 10.3 for an illustration. Each cell of ρ^* is either a cell of ρ' or of ρ_v and is thus $\mathcal{H}^{(k)}$-compatible. Therefore, ρ^* is $\mathcal{H}^{(k)}$-compatible, and thus G is $\mathcal{H}^{(k)}$-planar. This concludes this sketch of the proof.

Interestingly, this new irrelevant vertex technique (Lemma 10.2.15) can essentially be applied to any problem Π as long as we can prove the following:

1. there is a big enough flat wall W in the input graph G, and

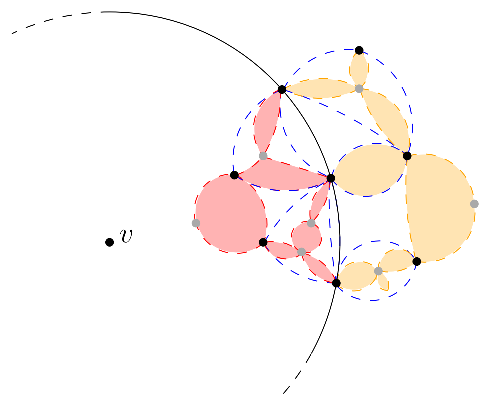

Figure 10.3: The black circle represents the boundary of Δ, ρ is represented in blue, ρ' in red, and ρ_v in orange.

2. there is a hereditary property $\mathcal{P}_\Pi$ such that Π can be expressed as the problem of finding a ground-maximal rendition ρ where each cell of ρ has the property $\mathcal{P}_\Pi$.

This is actually what we do for $\mathcal{H}$-planar treewidth, where, instead of asking the cells to be $\mathcal{H}$-compatible, we ask them to have another property $\Pi_{\mathcal{H},k}$ (see Subsection 10.4.1).

For item 2 above, there is actually also another constraint that was hidden under the carpet for this sketch, demanding that we should be able to choose ρ such that no cell of ρ contains W. In our case, this is what requires us to consider an embedding on the sphere instead of a disk to wisely choose the disks that will become $\mathcal{H}^{(k)}$-compatible cells, hence our use of sphere decompositions instead of renditions. Additionally, we crucially use the fact that a big wall is not contained in a component of bounded size, so the same argument would not work if we were to consider $\mathcal{H}$-Planarity in general instead of $\mathcal{H}^{(k)}$-Planarity (it could actually work as long as $\mathcal{H}$ is a class of bounded treewidth, given the parametric duality between walls and treewidth).

10.1.3 Changes for $\mathcal{H}$-planar treedepth

We sketch here Lemma 10.1.6, or more exactly, we explain the differences with Theorem 10.1.4. Contrary to the $\mathcal{H}^{(a)}$-planarity case, a big apex grid is not an obstruction for a graph of bounded $\mathcal{H}^{(a)}$-planar treedepth. Therefore, we cannot use the Flat Wall theorem variant of Theorem 10.2.2. Instead, we use the classical Flat Wall theorem (Proposition 4.6.3). If it reports that G has bounded treewidth, then we use Courcelle's theorem (Proposition 4.3.2) to conclude. If we obtain that G has a clique-minor of size $a + 4k$, then this is a no-instance. So we can assume that we find a small apex set A and a wall W that is flat in $G - A$ and whose compass has bounded treewidth. Let A^+ be the set of vertices of A with many disjoint paths to W and $A^- = A \setminus A^+$. Assuming G has a certifying elimination sequence $X_1, \ldots, X_k$, what we can essentially prove is that W is mostly part of X_i for some $i \in [k]$, and that the vertices of A^+ must belong to $\bigcup_{j<i} X_j$ and can hence be somewhat ignored. Hence, we do the following. We find a subwall W' of W that avoids the vertices of A^-, and thus is flat in $G - A^+$. We divide W' further in enough subwalls $W_1, \ldots, W_r$ (which are flat in $G - A^+$) so that we are sure that one of them, say W_p, is totally contained in X_i (though we do not know which one). Given that the compass of the walls W_ℓ have bounded treewidth, we can use Courcelle's theorem to compute the $\mathcal{H}^{(a)}$-planar treedepth d_ℓ of each of them. If, for each $\ell \in [r]$, $d_\ell > k$, then this is a no-instance. Otherwise, we chose ℓ such that d_ℓ is minimum. Then, given a central vertex v of $G - v$, we report a yes-instance if and only if $G - v$ has $\mathcal{H}$-planar treedepth at most k.

The argument for the irrelevancy of v use the core result as in the $\mathcal{H}^{(a)}$-planarity case, that is Lemma 10.2.15. What we essentially prove is that, given a certifying elimination sequence $Y_1, \dots, Y_k$ of $G - v$, most of W_ℓ is in Y_i, and that, as such, we can replace $Y_i, \dots, Y_k$ by according to the certifying elimination sequence we found for the compass of W_ℓ, to obtain a certifying elimination sequence $Y'_1, \dots, Y'_k$ of G'.

10.2 The FPT algorithm for $\mathcal{H}^{(k)}$-Planarity

In Subsection 10.2.1, we define *flat walls* and introduce the Flat Wall theorem. In Subsection 10.2.2, we provide an obstruction set to $\mathcal{H}^{(k)}$-Planarity. In Subsection 10.2.3, we observe that a yes-instance of $\mathcal{H}^{(k)}$-Planarity is essentially a graph with a special sphere decomposition, called $\mathcal{H}^{(k)}$*-compatible sphere decomposition*, and we introduce the notion of *ground-maximality* for sphere decompositions. In Subsection 10.2.4, we introduce the notion of *well-linkedness* and prove that a ground-maximal sphere decomposition is always *more grounded* than a well-linked sphere decomposition. Finally, in Subsection 10.2.6, we combine the results of the previous subsections together to prove Theorem 10.1.4.

10.2.1 Flat walls

The Flat Wall theorem [194, 271] essentially states that a graph G either contains a big clique as a minor or has bounded treewidth, or contains a flat wall after removing some vertices. We need here the version of the Flat Wall theorem of [286, Theorem 8].

Apex grid. The *apex grid* of height k is the graph Γ_k^+ obtained by adding a universal vertex to the $(k \times k)$-grid, i.e., a vertex adjacent to every vertex of $\Gamma_{k,k}$.

Proposition 10.2.1 (Lemma 3.1, [82])**.** *Let $m, k \in \mathbb{N}$ with $m \geq k^2 + 2k$ and let H be the $(m \times m)$-grid. Let X be a subset of at least k^4 vertices in the central $((m-2k) \times (m-2k))$-subgrid of H. Then there is a model of the $(k \times k)$-grid in H in which every branch set intersects X.*

The following result is a version of the Flat Wall theorem that is already somewhat known (see for instance in the proof of [205, Lemma 4.7]). We write here an algorithmic version that has yet to be stated, to our knowledge.

Theorem 10.2.2. *There exist a function $f_{10.2.2} : \mathbb{N} \to \mathbb{N}$ and an algorithm that, given a graph G and $k, r \in \mathbb{N}$ with r odd, outputs one of the following in time $2^{\mathcal{O}_k(r^2)} \cdot (n+m)$:*

- *a report that G contains an apex grid of height k as a minor,*
- *a report that $\mathsf{tw}(G) \leq f_{10.2.2}(k) \cdot r$, or*
- *a flatness pair $(W, \mathfrak{R})$ of G of height r whose $\mathfrak{R}$-compass has treewidth at most $f_{10.2.2}(k) \cdot r$.*

Moreover, $f_{10.2.2}(k) = 2^{\mathcal{O}(k^4 \log k)}$.

Proof. Let $t := k^2 + 1$, $a := g_{4.6.3}(t)$, $d := k^4$, $s := (d-1) \cdot (a-1) + 1$, $r_2 = \mathsf{odd}(\lceil \sqrt{s} \cdot (r+2) \rceil)$, and $r_1 := r_2 + 2k$.

We apply the algorithm of Proposition 4.6.3 that, in time $2^{\mathcal{O}_t(r_1^2)} \cdot n$, either reports that K_t is a minor of G, or finds a tree decomposition of G of width at most $f_{4.6.3}(t) \cdot r_1$, or finds a set $A \subseteq V(G)$ of size at most a, a flatness pair $(W_1, \mathfrak{R}_1)$ of $G - A$ of height r_1, and a tree decomposition of the

$\mathfrak{R}_1$-compass of W_1 of width at most $f_{4.6.3}(t) \cdot r_1$. In the first case, G thus contains an apex grid of order k as a minor, so we conclude. In the second case, we also immediately conclude. We can thus assume that we found a flatness pair $(W_1, \mathfrak{R}_1)$ of $G - A$. Let W_2 be the central r_2-subwall of W_1.

Given that $r_2 \geq \lceil \sqrt{s} \cdot (r+2) \rceil$, we can find a collection $\mathcal{W}' = \{W'_1, \ldots, W'_s\}$ of r-subwalls of W_2 such that $\mathsf{influence}_{\mathfrak{R}_1}(W'_i)$ and $\mathsf{influence}_{\mathfrak{R}_1}(W'_j)$ are disjoint for distinct $i, j \in [s']$. Then, by applying the algorithm of Proposition 4.6.6, in time $\mathcal{O}(n+m)$, we find a collection $\mathcal{W} = \{(W_1, \mathfrak{R}^1), \ldots, (W_s, \mathfrak{R}^s)\}$ such that, for $i \in [s]$, $(W_i, \mathfrak{R}^i)$ is a W'_i-tilt of $(W_1, \mathfrak{R}_1)$, and the $\mathfrak{R}^i$-compasses of the W_is are pairwise disjoint and have treewidth at most $f_{4.6.3}(t) \cdot r_1$.

For each vertex $v \in A$, we check whether v is adjacent to vertices of the compass of d subwalls in $\mathcal{W}$. If that is the case for some $v \in A$, then observe that G contains as a minor an $(r_1 \times r_1)$-grid (obtained by contracting the intersection of horizontal and vertical paths of W_1) along with a vertex (corresponding to v) that is adjacent to d vertices of its central $(r_2 \times r_2)$-subgrid (corresponding to W_2). But then, by Proposition 10.2.1, G contains an apex grid of height k as a minor, so we once again conclude.

We can thus assume that every vertex in A is adjacent to vertices of the compass of at most $d-1$ subwalls in $\mathcal{W}$. Given that $|\mathcal{W}| = s$ and that $|A| \leq a$, it implies that there is at least one wall W_i in $\mathcal{W}$ whose $\mathfrak{R}^i$-compass is adjacent to no vertex in A. Hence the result. □

10.2.2 An obstruction to $\mathcal{H}^{(k)}$-Planarity

In this section, we show that apex grids are obstructions to the existence of planar $\mathcal{H}^{(k)}$-modulators.

Lemma 10.2.3. *Let $\mathcal{H}$ be an arbitrary graph class and let k be a positive integer. Then any graph G containing the apex grid $\Gamma^+_{k'}$ for $k' \geq \sqrt{k+4}+2$ as a minor does not admit a planar $\mathcal{H}^{(k)}$-modulator.*

Proof. Let $\mathcal{G}$ be the class of all graphs. Because any graph class $\mathcal{H} \subseteq \mathcal{G}$, it is sufficient to show the lemma for $\mathcal{H} = \mathcal{G}$. For this, we prove the claim for $G = \Gamma^+_{k'}$.

Claim 10.2.4. *The apex grid $\Gamma^+_{k'}$ does not admit a planar $\mathcal{G}^{(k)}$-modulator.*

Proof of claim. The proof is by contradiction. Assume that $X \subseteq V(\Gamma^+_{k'})$ is a planar $\mathcal{G}^{(k)}$-modulator. Because $|V(\Gamma^+_{k'})| > k+5$, it implies that $X \neq \emptyset$, and because $\Gamma^+_{k'}$ is not planar as $k' \geq 3$, it implies that $V(\Gamma^+_{k'}) \setminus X \neq \emptyset$. Let v be the apex of $\Gamma^+_{k'}$. Denote by B the vertices of $\Gamma_{k'}$ of degree at most four, and set $S := V(\Gamma^+_{k'}) \setminus N_{\Gamma^+_{k'}}(B)$, that is S is the set of vertices of $\Gamma^+_{k'} - v$ that do not belong to the two outermost cycles of the grid. It is straightforward to verify that for any two distinct nonadjacent vertices $x, y \in S$, $\Gamma^+_{k'}$ has five internally vertex disjoint x-y-paths: four paths in $\Gamma^+_{k'} - v$ and one path with the middle vertex v. Therefore, for any separation (L, R) of $\Gamma^+_{k'}$ of order at most four with $L \setminus R \neq \emptyset$ and $R \setminus L \neq \emptyset$, either $S \subseteq L$ or $S \subseteq R$. Furthermore, because v is universal, $v \in L \cap R$. For each connected component C of $\Gamma^+_{k'} - X$, $(N_{\Gamma^+_{k'}}[V(C)], V(G) \setminus V(C))$ is a separation of order at most four. This implies that either $S \subseteq N_{\Gamma^+_{k'}}[V(C)]$ for a connected component C of $\Gamma^+_{k'} - X$ or $S \subseteq X$. However, because $|S| \geq k+4$ and $v \in N_{\Gamma^+_{k'}}(V(C))$, in the first case, we would have that $|V(C)| > k$ contradicting that each connected component of $\Gamma^+_{k'} - X$ has at most k vertices. Thus, $S \subseteq X$. We also have that $v \in X$. Then because $k' \geq 5$, $\Gamma^+_{k'}[S]$ contains $\Gamma_{3,3}$ as a subgraph and, therefore, $\Gamma^+_{k'}[S \cup \{v\}]$ is not planar. This contradicts that the torso of X is planar and proves the claim. ⋄

Given that $\mathcal{G}^{(k)}$ is a minor-closed graph class, so is the class of $\mathcal{G}^{(k)}$-planar graphs. Therefore, for any graph G containing $\Gamma^+_{k'}$ as a minor, G is not a $\mathcal{G}^{(k)}$-planar graph by Claim 10.2.4. This completes the proof. □

10.2.3 $\mathcal{H}$-compatible sphere decompositions

In this subsection, we observe that the problem of $\mathcal{H}$-PLANARITY has an equivalent definition using sphere decompositions.

$\mathcal{H}$-compatible sphere decompositions. Let $\mathcal{H}$ be a graph class. Let also G be a graph and $\delta = (\Gamma, \mathcal{D})$ be a sphere decomposition of G. We say that a cell c of δ is *$\mathcal{H}$-compatible* if there is a set $S_c \subseteq V(\sigma(c))$ containing $\pi_\delta(\tilde{c})$ such that $\mathsf{torso}(\sigma(c), S_c)$ has a planar embedding with the vertices of $\pi_\delta(\tilde{c})$ on the outer face and such that, for each $D \in \mathsf{cc}(\sigma(c) - S_c)$, $D \in \mathcal{H}$. We say that δ is *$\mathcal{H}$-compatible* if every cell of δ is $\mathcal{H}$-compatible. See Figure 10.1 for an illustration.

We show the following lemma.

Lemma 10.2.5. *Let $\mathcal{H}$ be a graph class, $k \in \mathbb{N}$, and G be a graph. Then G is $\mathcal{H}^{(k)}$-planar if and only if G has an $\mathcal{H}^{(k)}$-compatible sphere decomposition δ. Additionally, for any r-wall W of G with $r \geq \max\{\sqrt{(k+7)/2}+2, 7\}$, we can choose δ such that the $(r-2)$-central wall W' of W is grounded in δ.*

As a side note, the following proof can be easily adapted to prove that G is $\mathcal{H}$-planar if and only if G has an $\mathcal{H}$-compatible sphere decomposition δ. However, in this case, the wall W may be completely contained in a cell of δ.

Proof. Suppose that G has an $\mathcal{H}^{(k)}$-compatible sphere decomposition δ. Then, for each cell $c \in C(\delta)$, there is a set $S_c \subseteq V(\sigma(c))$ containing $\pi_\delta(\tilde{c})$ such that $\mathsf{torso}(\sigma(c), S_c)$ has a planar embedding with the vertices of $\pi_\delta(\tilde{c})$ on the outer face and such that, for each $D \in \mathsf{cc}(\sigma(c) - S_c)$, $D \in \mathcal{H}^{(k)}$. Then we immediately get that $\bigcup_{c \in C(\delta)} S_c$ is a planar $\mathcal{H}^{(k)}$-modulator of G. This comes from the fact that the disks in the sphere decomposition are disjoint apart from shared boundary vertices. Therefore, we can take the planar embeddings of the torsos of the individual cells that have the size-3 boundaries on the outer face, mirror them as needed to get the ordering along the boundary to match the ordering in the sphere decomposition, and then to use the cell-torso drawings into the sphere decomposition to get a complete drawing of the entire torso on the sphere, which is a planar drawing.

Suppose now that G is $\mathcal{H}^{(k)}$-planar. Let S be a planar $\mathcal{H}^{(k)}$-modulator in G. Let $V' \subseteq V(W')$ be set of 3-branch vertices of W that are vertices of W'. $|V'| = 2(r-2)^2 - 2 \geq k+5$. We first make the following observation.

Claim 10.2.6. *For any separation (A, B) of order at most three in G, the graph induced by one of A and B, say B, contains no cycle of W', and $B \setminus A$ contains at most one vertex of V'.*

Let $\delta = (\Gamma, \mathcal{D})$ be a sphere embedding of $\mathsf{torso}(G, S)$. Note that, for a sphere embedding, the set of its nodes contains all vertices of the drawing, since each disk of the sphere embedding surrounds a single edge and therefore no vertex of the drawing lies in the interior of any disk. For each δ-aligned disk Δ, let $Z_\Delta := \{C \in \mathsf{cc}(G - S) \mid N_G(V(C)) \subseteq \pi_\delta(N(\delta) \cap \Delta)\}$. Let also $V(Z_\Delta) := \bigcup_{C \in Z_\Delta} V(C)$ be the set of vertices of components in Z_Δ.

Claim 10.2.7. *Let $D \in \mathsf{cc}(G - S)$. There is a δ-aligned disk Δ_D such that*

- *the vertices of $N_G(V(D))$ are in the disk Δ_D, i.e. $N_G(V(D)) \subseteq \pi_\delta(N(\delta) \cap \Delta_D)$, with all but at most one (in the case $|N_G(V(D))| \leq 4$) being exactly the vertices of the boundary of Δ_D, i.e. there is a set $X_D \subseteq N_G(V(D))$ of size $\min\{|N_G(V(D))|, 3\}$ such that $X_D = \pi_\delta(\mathsf{bd}(\Delta_D) \cap N(\delta))$, and*
- *the graph induced by $B_D := V(\mathsf{inner}_\delta(\Delta_D)) \cup V(Z_\Delta)$ contains no cycle of W'.*

Proof of claim. Let $Y_D := \{C \in \mathsf{cc}(G - S) \mid N_G(V(C)) \subseteq N_G(V(D))\}$ and $V(Y_D) := \bigcup_{C \in Y_D} V(C)$ be the set of vertices of components in Y_D. We consider three cases depending on the size of $N_G(V(D))$.

Case 1: $|N_G(V(D))| \leq 2$. We set Δ_D to be

- the empty disk if $N_G(V(D)) = \emptyset$
- $\pi_\delta^{-1}(v)$ if $N_G(V(D)) = \{v\}$
- the closure of the cell $c \in C(\delta)$ such that $\sigma(c)$ is the edge induced by u, v in $\mathsf{torso}(G, S)$ if $N_G(V(D)) = \{u, v\}$.

Case 2: $|N_G(V(D))| = 3$. Informally, the triangle induced by $N_G(V(D))$ in $\mathsf{torso}(G, S)$ defines two disks Δ_1 and Δ_2 in δ. We need to show that one of them is the desired disk, that does not contain a cycle of W'.

Let T be the cycle in the embedding δ induced by $N_G(V(D))$. $\mathbb{S}^2 \setminus T$ is the union of two open disks whose closure is respectively called Δ_1 and Δ_2. For $i \in [2]$, let $A_i := V(\mathsf{inner}_\delta(\Delta_i)) \cup V(Z_{\Delta_i}) \setminus V(Y_D)$. Then (A_1, A_2) is a separation of $G - V(Y_D)$ with $A_1 \cap A_2 = N_G(V(D))$. By Claim 10.2.6, the graph induced by one side of the separation, say A_2 contains no cycle of W'. We set Δ_D to be a δ-aligned disk containing Δ_1.

It remains to prove that $G[B_D]$ contains no cycle of W', where $B_D := V(\mathsf{inner}_\delta(\Delta_D)) \cup V(Z_\Delta) = A_2 \cup V(Y_D)$. (A_1, B_D) is a separation of G with $A_1 \cap B_D = N_G(V(D))$, so by Claim 10.2.6, one of A_1 and B_D induce a graph containing no cycle of W'. Assume towards a contradiction that $G[A_1]$ contains no cycle of W'. Note that, given that $r - 2 \geq 5$, W' contains a set $\mathcal{C}$ of pairwise disjoint cycles with $|\mathcal{C}| \geq 4$. Given that $|A_1 \cap A_2| = |N_G(V(D))| = 3$, at most three cycles of $\mathcal{C}$ intersects $N_G(V(D))$ and $G[A_1 \cup A_2]$ contains at most one cycle of $\mathcal{C}$. Therefore, Y_D contains at least one cycle of $\mathcal{C}$. Let $C \in Y_D$ by a component containing such a cycle. We have that $(V(C) \cup N_G(V(C)), V(G) \setminus V(C))$ is a separation of order at most three so, given that C contains a cycle of W', by Claim 10.2.6, C contains at least $|V'| - 4 > k$ vertices of V'. This contradicts the fact that $|V(C)| \geq k$.

Case 3: $|N_G(V(D))| = 4$. Let $\{v_i \mid i \in [4]\}$ be the vertices in $N_G(V(D))$ and $X_i := N_G(V(D)) \setminus \{v_i\}$. For $i \in [4]$, we define Δ_i similarly to Δ_D in the previous case. Then $X_i = \pi_\delta(\mathsf{bd}(\Delta_D) \cap N(\delta))$ and the graph induced by $B_i := V(\mathsf{inner}_\delta(\Delta_i)) \cup V(Z_{\Delta_i})$ contains no cycle of W'. However, it might be the case that $v_i \notin \pi_\delta(N(\delta) \cap \Delta_i)$.

If $v_i \notin \pi_\delta(N(\delta) \cap \Delta_i)$ for all $i \in [4]$, then the interior of the Δ_i are pairwise disjoint. Moreover, $\bigcup_{i \in [4]} B_i = V(G)$, $\bigcap_{i \in [4]} B_i = V(Y_D) \cup N_G(V(D))$, and $N_G(V(Y_D)) \subseteq N_G(V(D))$. As shown in the previous case, $G[B_i]$ contains no cycle of W' for $i \in [4]$. This implies that, for any cycle of W', there are distinct $i, j \in [4]$ such that the cycle has a vertex in $B_i \setminus B_j$ and $B_j \setminus B_i$, and thus intersects $N_G(V(D))$ twice. However, W' has at least three pairwise disjoint cycles, a contradiction to the fact that $|N_G(V(D))| = 4$. Therefore, there is $i \in [4]$ such that $v_i \in \pi_\delta(N(\delta) \cap \Delta_i)$. We then set $\Delta_D := \Delta_i$. This completes the case analysis and the proof of the claim. $\diamond$

Note that, if $N_G(V(D)) = N_G(V(D'))$, then we can assume that $\Delta_D = \Delta_{D'}$. Let $\mathcal{D}^*$ be the inclusion-wise maximal elements of $\mathcal{D} \cup \{\Delta_D \mid D \in \mathsf{cc}(G - S)\}$. By maximality of $\mathcal{D}^*$ and planarity of $\mathsf{torso}(G, S)$, any two distinct $\Delta_D, \Delta_{D'} \in \mathcal{D}^*$ may only intersect on their boundary. For each $C \in \mathsf{cc}(G - S)$, we draw C in a $\Delta_D \in \mathcal{D}$ such that $V(C) \subseteq B_D$, and add the appropriate edges with $\pi_\delta(N(\delta) \cap \mathsf{bd}(\Delta_D))$. We similarly draw the edges of $G[S]$ to obtain a drawing Γ^* of G.

For each cell c of $\delta^* = (\Gamma^*, \mathcal{D}^*)$, there is $D \in \mathsf{cc}(G - S)$ such that $\sigma_{\delta^*}(c)$ contains no cycle of W', since $\sigma_{\delta^*}(c)$ is a subgraph of $G[B_D]$, so W' is grounded in δ^*. Moreover, $S_c := V(\sigma(c)) \cap V(\mathsf{torso}(G, S))$ certifies that c is $\mathcal{H}^{(k)}$-compatible. $\square$

Moreover, if $\mathcal{H}$ is a hereditary graph class, then we can "ground" an $\mathcal{H}$-compatible sphere decomposition as much as possible. This is what we prove in Lemma 10.2.8 after defining the relevant definitions.

Containment of cells. Let $\delta = (\Gamma, \mathcal{D})$ and $\delta' = (\Gamma', \mathcal{D}')$ be two sphere decompositions of G. Let $c \in C(\delta)$ and $c' \in C(\delta')$ be two cells. We say that c *is contained in* c' if $V(\sigma(c)) \subseteq V(\sigma_{\delta'}(c'))$. We say that c *and* c' *are equivalent* if c is contained in c' and c' is contained in c.

Ground-maximal sphere decompositions. Let $\delta = (\Gamma, \mathcal{D})$ and $\delta' = (\Gamma', \mathcal{D}')$ be two sphere decompositions of G. We say that δ is *more grounded* than δ' (and that δ' is *less grounded* than δ) if each cell $c \in C(\delta)$ is contained in a cell $c' \in C(\delta')$, and in case c and c' are equivalent, if $\pi_{\delta'}(\tilde{c}') \subseteq \pi_\delta(\tilde{c})$. We say that δ is *ground-maximal* if no other sphere decomposition of G is more grounded than δ. We say that a cell $c \in C(\delta)$ is *ground-maximal* if, for any sphere decomposition $\delta' = (\Gamma', \mathcal{D}')$ that is more grounded than δ and for any cell $c' \in C(\delta')$ that is contained in c, $V(\sigma(c)) = V(\sigma_{\delta'}(c'))$ and $\pi_{\delta'}(\tilde{c}') = \pi_\delta(\tilde{c})$.

Lemma 10.2.8. *Let $\mathcal{H}$ be a hereditary graph class and G be a graph. Let δ be a sphere decomposition of G that is $\mathcal{H}$-compatible. Then any sphere decomposition of G that is more grounded than δ is also $\mathcal{H}$-compatible.*

Proof. Let $\delta' = (\Gamma', \mathcal{D}')$ be a sphere decomposition that is more grounded than $\delta = (\Gamma, \mathcal{D})$. Let $c' \in C(\delta')$. Let us show that c' is $\mathcal{H}$-compatible. Given that δ' is more grounded than δ, there is a cell $c \in C(\delta)$ such that $V(\sigma_{\delta'}(c')) \subseteq V(\sigma(c))$.

Let $U = V(\sigma(c)) \setminus V(\sigma_{\delta'}(c'))$. Given that c is $\mathcal{H}$-compatible, there is a set $S_c \subseteq V(\sigma(c))$ containing $\pi_\delta(\tilde{c})$ such that $\mathsf{torso}(\sigma(c), S_c)$ has a planar embedding with the vertices of $\pi_\delta(\tilde{c})$ on the outer face and such that, for each $D \in \mathsf{cc}(\sigma(c) - S_c)$, $D \in \mathcal{H}$. Let $S_{c'} := S_c \setminus U$. By heredity of $\mathcal{H}$, each connected component C of $\sigma(c) - S_c - U = \sigma_{\delta'}(c') - S_{c'}$ belong to $\mathcal{H}$. Additionally, we have $\mathsf{torso}(\sigma_{\delta'}(c'), S_{c'}) = \mathsf{torso}(\sigma(c) - U, S_c \setminus U) = \mathsf{torso}(\sigma(c), S_C) - U$, so $\mathsf{torso}(\sigma_{\delta'}(c'), S_{c'})$ is planar. The result follows. $\square$

Combining Lemma 10.2.5 and Lemma 10.2.8, we obtain the following result.

Corollary 10.2.9. *Let $\mathcal{H}$ be a hereditary graph class, $k \in \mathbb{N}$, G be a graph, and W be an r-wall in G with $r \geq \max\{\sqrt{(k+7)/2} + 2, 7\}$. A graph G is $\mathcal{H}^{(k)}$-planar if and only if G has a ground-maximal $\mathcal{H}^{(k)}$-compatible sphere decomposition δ. Additionally, we can choose δ such that the $(r-2)$-central wall of W is grounded in δ.*

10.2.4 Comparing sphere decompositions

In this subsection, we essentially want to prove that, given two sphere decompositions δ_1 and δ_2 on the same graph, if δ_1 is *ground-maximal* and δ_2 is *well-linked*, then δ_1 is always *more grounded* than δ_2 (see Corollary 10.2.14).

Well-linkedness. Let $\rho = (\Gamma, \mathcal{D})$ be a rendition of (G, Ω). We say that a cell $c \in C(\delta)$ is *well-linked* if there are $|\tilde{c}|$ vertex-disjoint paths from $\pi_\delta(\tilde{c})$ to $V(\Omega)$. We say that δ is *well-linked* if every cell $c \in C(\delta)$ is well-linked.

The following result is a corollary of [286].

Proposition 10.2.10 (Lemma 3, [286])**.** *If a society (G, Ω) has a rendition, then (G, Ω) has a well-linked rendition.*

Intersection and crossing of cells. Let $\delta = (\Gamma, \mathcal{D})$ and $\delta' = (\Gamma', \mathcal{D}')$ be two sphere decompositions of a graph G. Let $c \in C(\delta)$ and $c' \in C(\delta')$ be two cells. We say that *c and c' intersect* if $(V(\sigma(c)) \cap V(\sigma_{\delta'}(c'))) \setminus (\pi_\delta(\tilde{c}) \cap \pi_{\delta'}(\tilde{c}')) \neq \emptyset$. We say that *$c$ and c' cross* if c and c' intersect but that neither of them is contained in the other.

Our goal in this subsection is to prove that if c is a ground-maximal cell in some sphere decomposition δ and that c' is some well-linked cell in some rendition δ', such that c and c' intersect, then c is contained in c' (see Lemma 10.2.13). We first define a splitting operation on a sphere decomposition that will be extensively used in this subsection.

Splitting a cell. Let $\delta = (\Gamma, \mathcal{D})$ and $\delta' = (\Gamma', \mathcal{D}')$ be two sphere decompositions of a graph G. Let $c \in C(\delta)$ be a cell and $v \in V(\sigma(c))$ be a vertex such that at least two connected components of $\sigma(c) - v$ contains a vertex of $\pi_\delta(\tilde{c})$. Such a vertex v is called a *cut-vertex of c in δ*. We say that δ^* is *obtained from δ by splitting c at v* if δ' can be constructed from $\delta - (V(\sigma(c)) \setminus \pi_{\delta(\tilde{c})}) = (\Gamma', \mathcal{D}')$ as follows.

Let C_1 be a disjoint union of components of $\sigma(c) - v$ such that C_1 and $C_2 := \sigma(c) - v - V(C_1)$ are both non-empty. Let $A_1 = V(C_1) \cup \{v\}$ and $A_2 = V(\sigma(c)) \setminus V(C_1)$. For $i \in [2]$, let $B_i = A_i \cap \pi_\delta(\tilde{c})$. Note that $1 \leq |B_i| \leq 2$.

We set $N(\delta^*) = N(\delta') \cup \{x\}$ for some arbitrary point x contained in c. We set $\mathcal{D}^* = \mathcal{D}' \cup \{\Delta_{c_1}, \Delta_{c_2}\}$, where $c_1, c_2 \subseteq c$ be two new cells such that $\tilde{c}_1 = \{x\} \cup \pi_\delta^{-1}(B_1)$ and $\tilde{c}_2 = \{x\} \cup \pi_\delta^{-1}(B_2)$. Then Γ^* is obtained from Γ' by arbitrarily drawing $G[A_i]$ in c_i, for $i \in [2]$. See Figure 10.4 for an illustration.

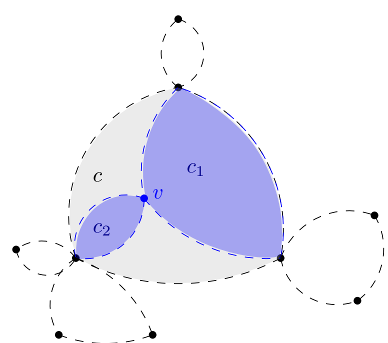

Figure 10.4: The blue sphere decomposition is obtained from the gray one by splitting c at v.

Observe that δ' is well-defined and that it is more grounded than δ. Thus, c is not ground-maximal.

In the following lemma, we prove that if two cells c and c' cross and that c' has exactly one vertex of its boundary in c, then c is not ground-maximal.

Lemma 10.2.11. *Let $\delta = (\Gamma, \mathcal{D})$ and $\delta' = (\Gamma', \mathcal{D}')$ be two sphere decompositions of a graph G. Let $c \in C(\delta)$ and $c' \in C(\delta')$ be two cells that cross and suppose that $v \in V(G)$ is the unique vertex of $\pi_{\delta'}(\tilde{c}')$ in $\sigma(c) - \pi_\delta(\tilde{c})$. Then v is a cut-vertex of c in δ, and therefore, c is not ground-maximal.*

Proof. Let $A = \pi_\delta(\tilde{c}) \setminus V(\sigma_{\delta'}(c')) \neq \emptyset$ and $A' = \pi_\delta(\tilde{c}) \cap V(\sigma_{\delta'}(c')) \neq \emptyset$. Let P be an $(A - A')$-path in $V(\sigma(c))$ (it exists by the connectivity of $\sigma(c)$). Given that one endpoint of P is in $V(\sigma_{\delta'}(c'))$ and that the other is not, it implies that P intersects $\pi_{\delta'}(\tilde{c}')$. Given that v is the unique vertex of $\pi_{\delta'}(\tilde{c}')$ in $V(\sigma(c))$, it implies that $v \in V(P)$ for all $(A - A')$-paths P. We conclude that at least two connected components of $\sigma(c) - v$ ocontains a vertex of $\pi_\delta(\tilde{c})$, and thus, v is a cut-vertex of c. $\square$

We now prove that a ground-maximal cell and a well-linked cell cannot cross.

Lemma 10.2.12. *Let (G, Ω) be a society. Let δ be a sphere decomposition of G and let δ' be a rendition of (G, Ω). Let $c \in C(\delta)$ be a ground-maximal cell such that $V(\Omega) \cap (V(\sigma(c)) \setminus \pi_\delta(\tilde{c})) = \emptyset$ and $c' \in C(\delta')$ be a well-linked cell. Then c and c' do not cross.*

Proof. Assume towards a contradiction that c and c' cross. Therefore, by connectivity of $\sigma(c)$ and $\sigma_{\delta'}(c')$, there is at least one vertex of $\pi_\delta(\tilde{c})$ in $V(\sigma_{\delta'}(c')) \setminus \pi_{\delta'}(\tilde{c}')$ and at least one in $V(\sigma(c)) \setminus V(\sigma_{\delta'}(c'))$. Similarly, there is at least one vertex of $\pi_{\delta'}(\tilde{c}')$ in $V(\sigma(c)) \setminus \pi_\delta(\tilde{c})$ and at least one in $V(\sigma_{\delta'}(c')) \setminus V(\sigma(c))$. Given that $|\tilde{c}| \leq 3$ and $|\tilde{c}'| \leq 3$, we can distinguish two cases:

- **Case 1:** there is exactly one vertex of $\pi_{\delta'}(\tilde{c}')$ in $\sigma(c) - \pi_\delta(\tilde{c})$ (Figure 10.5).
- **Case 2:** there are exactly two vertices of $\pi_{\delta'}(\tilde{c}')$ in $\sigma(c) - \pi_\delta(\tilde{c})$ (Figure 10.6).

Assume first that Case 1 happens. Let v be the unique vertex of $\pi_{\delta'}(\tilde{c}')$ in $\sigma(c) - \pi_\delta(\tilde{c})$. By

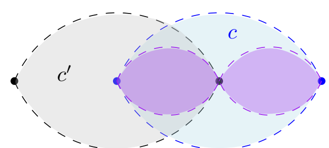

Figure 10.5: Case 1 of Lemma 10.2.12.

Lemma 10.2.11, v is a cut-vertex of c in δ, and therefore c is not ground-maximal. Hence, Case 1 does not apply by maximality of c.

Assume that Case 2 applies. Let a_1 and a_2 be the vertices of $\pi_{\delta'}(\tilde{c}')$ in $\sigma(c) - \pi_\delta(\tilde{c})$. Let

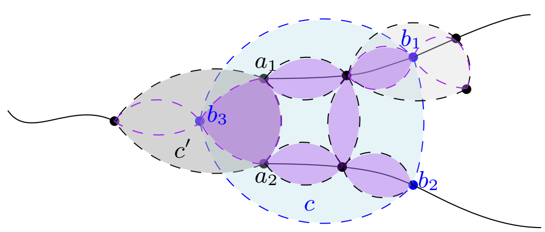

Figure 10.6: Case 2 of Lemma 10.2.12.

$B = \pi_\delta(\tilde{c}) \setminus V(\sigma_{\delta'}(c'))$ and $B' = \pi_\delta(\tilde{c}) \cap V(\sigma_{\delta'}(c'))$. Remember that $1 \leq |B|, |B'| \leq 2$. Given that c' is well-linked, there are two vertex-disjoint paths P_1 and P_2 in $G - V(\sigma(c'))$, the first one from a_1 to $V(\Omega)$ and the other from a_2 to $V(\Omega)$. Given that $a_1, a_2 \in V(\sigma(c))$ and that $V(\Omega) \cap (V(\sigma(c)) \setminus \pi_\delta(\tilde{c})) = \emptyset$, it implies that P_1 and P_2 intersect B, say in b_1 and b_2 respectively. Given that $B = \{b_1, b_2\}$, we thus have $|B'| = 1$. By Lemma 10.2.11, the unique vertex b_3 in B' is a cut-vertex for c' in δ'. Given that every edge of G is contained in a cell of δ', for $i \in [2]$, there is a cell $c_i \in C(\delta')$ containing the edge $b_i'b_i$ where b_i' is the neighbor of b_i in $V(P_i) \cap V(\sigma(c))$.

Observe that c_1 and c_2 are distinct cells since otherwise we would have $|\pi_{\delta'}(\tilde{c_1})| \geq |\pi_{\delta'}(\tilde{c_1}) \cap V(P_1)| + |\pi_{\delta'}(\tilde{c_1}) \cap V(P_2)| \geq 4$. Therefore, for $i \in [2]$, either $V(\sigma_{\delta'}(c_i)) \subseteq V(\sigma(c))$, or by Lemma 10.2.11, b_i

is a cut-vertex for c_i in δ'. Let δ'' be the rendition obtained from δ' by splitting c' at b_3, c_1 at b_1, and c_2 at b_2 (only if those are cut-vertices for the last two cases). Note that δ'' is more grounded that δ' and that no cell $c'' \in C(\delta'')$ crosses c. Let $\mathcal{C} := \{c'' \in C(\delta'') \mid c''$ contained in $c\}$ and Δ'' be a δ''-aligned disk containing exactly the cells of $\mathcal{C}$. Up to homeomorphism, we may assume that $\Delta'' = \Delta_c$ and that $\pi_{\delta''}(v) = \pi_\delta(v)$ for all $v \in \mathsf{bd}(\Delta'') \cap N(\delta'') = \mathsf{bd}(\Delta_c) \cap N(\delta)$. Then we define δ^* to be the sphere decomposition of G that is equal to δ'' when restricted to c, and equal to δ otherwise. δ^* is more grounded than δ, and, in particular, c is not ground-maximal. Hence, Case 2 does not apply by maximality of c. This contradiction concludes the proof. □

Lemma 10.2.13. *Let (G, Ω) be a graph. Let $\delta = (\Gamma, \mathcal{D})$ be a sphere decomposition of G and let $\delta' = (\Gamma', \mathcal{D}')$ be a rendition of (G, Ω). Let $c \in C(\delta)$ be a ground-maximal cell such that $V(\Omega) \cap (V(\sigma(c)) \setminus \pi_\delta(\tilde{c})) = \emptyset$. Suppose that every cell $c' \in C(\delta')$ that intersects c is well-linked. Then c is contained in a cell $c' \in C(\delta')$.*

Proof. By Lemma 10.2.12, no well-linked cell $c' \in C(\delta')$ crosses c. Hence, every cell $c' \in C(\delta')$ that intersects c is either contained in c or contains c. If c is contained in c', we can immediately conclude, so let us assume that every cell $c' \in C(\delta')$ that intersects c is contained in c. We define $\delta^* = (\Gamma^*, \mathcal{D}^*)$ to be the rendition of (G, Ω) that is equal to δ' when restricted to c, and equal to δ otherwise, similarly to δ^* in the Case 2 of Lemma 10.2.12 Thus, δ^* is more grounded than δ. Hence, given that c is ground-maximal, we conclude that, for any $c^* \in C(\delta^*)$ contained in c, we have $V(\sigma(c)) = V(\sigma_{\delta^*}(c^*))$. But, for any $c^* \in C(\delta^*)$ contained in c, there is $c' \in C(\delta')$ such that $V(\sigma_{\delta^*}(c^*)) = V(\sigma_{\delta'}(c'))$. Therefore, c is contained in a cell $c' \in C(\delta')$. □

Given that, if δ is a rendition of a society (G, Ω), then $V(\Omega) \subseteq \pi_\delta(N(\delta))$, and thus $V(\Omega) \cap (V(\sigma(c)) \setminus \pi_\delta(\tilde{c})) = \emptyset$ for any cells in δ, we immediately get the following corollary from Lemma 10.2.13.

Corollary 10.2.14. *Let G be a connected graph. Let δ be a ground-maximal rendition of (G, Ω) and let δ' be a well-linked rendition of (G, Ω). Then δ is more grounded than δ'.*

10.2.5 Combining sphere decompositions

To prove Theorem 10.1.4, as well as Lemma 10.1.6 and Lemma 10.1.7 later, the main ingredient is the following result that says that, given a ground-maximal sphere decomposition of the compass of some flat wall W of G and given a ground-maximal sphere decomposition of $G - Y$, where Y is a central part of W, these two sphere decompositions can be glued to obtain a sphere decomposition of G.

Lemma 10.2.15. *Let $k, r, q \in \mathbb{N}$ with r, q odd and $r \geq q + 10$. Let G be a graph, $(W, \mathfrak{R} = (A, B, P, C, \delta))$ be a flatness pair of G of height r, G' be the $\mathfrak{R}$-compass of W, and Y be the vertex set of the $\mathfrak{R}^{(q)}$-compass of a $W^{(q)}$-tilt $(\tilde{W}^{(q)}, \mathfrak{R}^{(q)})$ of $(W, \mathfrak{R})$. Suppose also that:*

- *$\delta = (\Gamma, \mathcal{D})$ is a well-linked rendition of (G', Ω), where Ω is the cyclic ordering of the vertices of $A \cap B$ as they appear in $D(W)$, and*
- *there are two ground-maximal sphere decompositions $\delta' = (\Gamma', \mathcal{D}')$ and $\delta_Y = (\Gamma_Y, \mathcal{D}_Y)$ of G' and $G - Y$, respectively, such that the $(r-2)$-central wall W' of W is grounded in both δ' and δ_Y.*

Let T be the track of third layer C_3 of W with respect to δ. Then $\pi_\delta(T) \subseteq N(\delta') \cap N(\delta_Y)$.

Consequently, there is a ground-maximal sphere decomposition δ^* of G such that each cell of δ^* is either a cell of δ' or a cell of δ_Y, and more specifically:

- *T is the track of C_3 with respect to δ^*,*
- *the restriction of δ^* to the closed disk delimited by T containing Y (resp.* not *containing Y) is, up to homeomorphism, the restriction of δ' (resp. δ_Y) to the closed disk delimited by T containing Y (resp.* not *containing Y).*

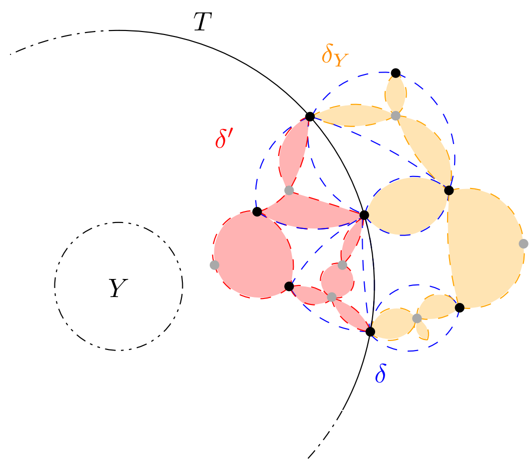

Figure 10.7: The black circle represents T, δ is represented in blue, δ' in red, and δ_Y in orange. δ^* is obtained by combining δ' inside of T and δ_Y outside of T.

Proof. Let C_i be the i-th layer of W for $i \in [(r-1)/2]$ (so C_1 is the perimeter of W). Let T be the track of C_3 in δ. Informally, we want to prove that near T, the cells of δ' and δ_Y are contained in cells of δ. Hence, we may replace δ by δ' in the disk bounded by T containing v, and we may replace δ by δ_Y in the disk bounded by T not containing v. See Figure 10.7 for an illustration.

To prove that each cell c of δ' near T is contained in a cell of δ, we want to apply Lemma 10.2.13. To do so, we need to check that the interior of c does not contain any vertex of $V(\Omega)$. Here, "near" T means intersecting T in at least two points, that is $|V(\sigma_{\delta'}(c)) \cap \pi_\delta(T)| \geq 2$. This is what we prove in Claim 10.2.16. We also prove it for δ_Y at the same time.

Claim 10.2.16. *Let $\delta_1 \in \{\delta', \delta_Y\}$. For each cell $c_1 \in C(\delta_1)$ such that $|V(\sigma_{\delta_1}(c_1)) \cap \pi_\delta(T)| \geq 2$, it holds that $V(\Omega) \cap V(\sigma_{\delta_1}(c_1)) = \emptyset$.*

Proof of claim. Let $c_1 \in C(\delta_1)$ be such that $V(\sigma_{\delta_1}(c_1)) \cap \pi_\delta(T) \neq \emptyset$. If $V(\Omega) \cap V(\sigma_{\delta_1}(c_1)) \neq \emptyset$, then $\sigma_{\delta_1}(c_1)$ contains vertices of both C_1 and C_3, and thus also of C_2. Given that W' is grounded in δ_1, it holds that $\sigma_{\delta_1}(c_1)$ does not contain C_2 nor C_3. Therefore, $\pi_\delta(\tilde{c}_1)$ contains two vertices of C_2. But given that C_2 and C_3 are vertex disjoint and that $\pi_\delta(\tilde{c}_1)$ contains two vertices of $\pi_\delta(T) \subseteq v(C_3)$, this implies that $|\tilde{c}_1| \geq 4$, a contradiction. Therefore, $V(\Omega) \cap V(\sigma_{\delta_1}(c_1)) = \emptyset$. This completes the proof. $\diamond$

Therefore, given that δ and δ' are sphere decomposition of the same graph G', we prove using Lemma 10.2.13 that any cell of δ' intersecting T in at least two point is contained in a cell of δ.

Claim 10.2.17. *For any $c' \in C(\delta')$ such that $|V(\sigma_{\delta'}(c')) \cap \pi_\delta(T)| \geq 2$, there is a cell $c \in C(\delta)$ such that c' is contained in c.*

Proof of claim. Given that δ is well-linked, that δ' is ground-maximal, and that $V(\Omega) \cap V(\sigma_{\delta'}(c')) = \emptyset$ by Claim 10.2.16, the proof immediately follows from Lemma 10.2.13. $\diamond$

For δ_Y however, using Lemma 10.2.13 is not that easy. Indeed, δ and δ_Y are not sphere decompositions of the same graph, given that one is a sphere decomposition of G' and the other of $G - Y$. Therefore, we need to consider the restrictions $\tilde{\delta}$ and $\tilde{\delta}_Y$ of δ and δ_Y to $G' - Y$ and to apply Lemma 10.2.13 to these sphere decompositions. To do so, we need to check that a cell c_Y of δ_Y near T is still a cell of $\tilde{\delta}_Y$, that is ground-maximal, and that a cell c of δ intersecting c_Y is still a cell of $\tilde{\delta}$, that is well-linked. Let us first prove that the vertices of Y are not vertices of $\sigma(c)$, and thus that c is still a cell of $\tilde{\delta}$.

Claim 10.2.18. *Let $c_Y \in C(\delta_Y)$ be such that $|V(\sigma_{\delta_Y}(c_Y)) \cap \pi_\delta(T)| \geq 2$. Then, for any $c \in C(\delta)$ that intersect c_Y, $Y \cap V(\sigma(c)) = \emptyset$.*

Proof of claim. Suppose towards a contradiction that $Y \cap V(\sigma(c)) \neq \emptyset$. The perimeter of $W^{(q)}$ is C_j for $j = (r-q)/2+1$. Given that $r \geq q+10$, this implies that $j \geq 6$. Hence, by definition of a tilt of a flatness pair, we conclude that $Y \cap V(C_5) = \emptyset$. Then, by connectivity, either $\sigma(c)$ or $\sigma_{\delta_Y}(c_Y)$ intersects C_4 and C_5. Given that W' is grounded in both δ and δ_Y, neither C_5 nor C_4 is totally contained in $\sigma(c)$ or $\sigma_{\delta_Y}(c_Y)$.

Given that $|\pi_{\delta_Y}(\tilde{c_Y}) \cap \pi_\delta(T)| \geq 2$ and that $|\pi_{\delta_Y}(\tilde{c_Y})| \leq 3$, C_4 cannot intersect $\pi_{\delta_Y}(\tilde{c_Y})$ in two vertices. Therefore, $\pi_\delta(c)$ must intersect both C_4 and C_5. Hence $|\tilde{c}| \geq 4$, a contradiction proving the claim. ◇

Let us now prove that any cell of δ_Y intersecting T in at least two points is contained in a cell of δ.

Claim 10.2.19. *For any $c_Y \in C(\delta_Y)$ such that $|V(\sigma_{\delta_Y}(c_Y)) \cap \pi_\delta(T)| \geq 2$, there is a cell $c \in C(\delta)$ such that c_Y is contained in c.*

Proof of claim. Let $\tilde{\delta} = (\tilde{\Gamma}, \tilde{\mathcal{D}})$ and $\tilde{\delta}_Y = (\tilde{\Gamma}_Y, \tilde{\mathcal{D}}_Y)$ be the restrictions of δ and δ_Y, respectively, to $G'' := G' - Y$, i.e., $\tilde{\delta} = \delta - Y$ and $\tilde{\delta}_Y = \delta_Y - (V(G) \setminus V(G'))$.

Let $c \in C(\delta)$ be a cell that intersects c_Y. By Claim 10.2.18, $Y \cap V(\sigma(c)) = \emptyset$, so $c \in C(\tilde{\delta})$. Let us show that c is still well-linked in $\tilde{\delta}$. Given that c is well-linked in δ, there are three vertex-disjoint paths from $\pi_\delta(\tilde{c})$ to $V(\Omega)$. Even if one of them contains a vertex of Y, we can reroute the paths using the wall so that they avoid Y. Hence, c is still well-linked after removing v.

Also, by Claim 10.2.16, $V(\Omega) \cap V(\sigma_{\delta_Y}(c_Y)) = \emptyset$, so $c_Y \in C(\tilde{\delta}_Y)$. Therefore, c_Y is ground-maximal in $\tilde{\delta}_Y$. Indeed, otherwise, there is a sphere decomposition $\tilde{\delta}'_Y$ of G'' that is more grounded than $\tilde{\delta}_Y$ such that, for any cell c'_Y in $\tilde{\delta}'_Y$, c'_Y is either contained in c_Y, or is equal to another cell of $\tilde{\delta}_Y$. But then we can easily add Y back to obtain a sphere decomposition δ'_Y that is more grounded than δ_Y, contradicting the maximality of δ_Y.

Hence, the proof follows from Lemma 10.2.13 applied on (G'', Ω). This completes the proof of the claim. ◇

Given that the cells of δ' and δ_Y near T are contained in cells of δ, it implies that no cell of δ' nor δ_Y intersects T. This is what we prove in Claim 10.2.20.

Claim 10.2.20. $\pi_\delta(T) \subseteq N(\delta') \cap N(\delta_Y)$.

Proof of claim. Let $\delta_1 \in \{\delta', \delta_Y\}$. Suppose towards a contradiction that $\pi_\delta(T) \setminus N(\delta_1) \neq \emptyset$. Then there is a cell $c_1 \in C(\delta_1)$ and a vertex $x \in \pi_\delta(T)$ such that $x \in V(\sigma_{\delta_1}(c_1)) \setminus \pi_{\delta_1}(\tilde{c}_1)$. Therefore, $|V(\sigma_{\delta_Y}(c_Y)) \cap \pi_\delta(T)| \geq 2$. Hence, by one of Claim 10.2.17 and Claim 10.2.19, there exists a cell $c \in C(\delta)$ such that c_1 is contained in c. However, given that T is a track in δ, this implies that $x \notin V(\sigma_\delta(c)) \setminus \pi_\delta(\tilde{c}) \supseteq V(\sigma_{\delta_1}(c_1)) \setminus \pi_{\delta_1}(\tilde{c}_1)$, a contradiction proving the claim. ◇

Therefore, there exists a δ'-aligned disk Δ' such that $\mathsf{inner}_\delta(C_3) = \mathsf{inner}_{\delta'}(\Delta')$, and a δ_Y-aligned disk Δ_v such that $\mathsf{inner}_\delta(C_3) - v = \mathsf{outer}_{\delta_Y}(\Delta_v)$. $\mathbb{S}^2 - T$ is composed of two open disks. Up to homeomorphism, we may assume that the closures of these disks are Δ_v and Δ', respectively. Hence, we can define $\delta^* = (\Gamma^*, \mathcal{D}^*)$ such that $\mathcal{D}^* := \{\Delta_c \in \mathcal{D}' \mid c \subseteq \Delta'\} \cup \{\Delta_c \in \mathcal{D}_v \mid c \subseteq \Delta_v\}$ and $\Gamma^* = (\Gamma' \cap \Delta') \cup (\Gamma_v \cap \Delta_v)$. Each cell c^* of δ^* is either a cell of δ' or of δ_Y, and is thus ground-maximal, hence the result. $\square$

A direct corollary is that, if the compass of some flat wall W of G is $\mathcal{H}^{(k)}$-planar and that $G - v$ is $\mathcal{H}^{(k)}$-planar, where v is the cental vertex of W, then G is $\mathcal{H}^{(k)}$-planar.

Corollary 10.2.21. *Let $\mathcal{H}$ be a hereditary graph class. Let $k, q, r \in \mathbb{N}$ with q, r odd and $r \geq \max\{\sqrt{(k+7)/2} + 2, q + 10\}$. Let G be a graph, $(W, \mathfrak{R} = (A, B, P, C, \rho))$ be a flatness pair of G of height r, G' be the $\mathfrak{R}$-compass of W, and Y be the vertex set of the $\mathfrak{R}^{(q)}$-compass of a $W^{(q)}$-tilt $(\tilde{W}^{(q)}, \mathfrak{R}^{(q)})$ of $(W, \mathfrak{R})$. Then G is a* yes*-instance of $\mathcal{H}^{(k)}$-*PLANARITY *if and only if G' and $G - Y$ are both* yes*-instances of $\mathcal{H}^{(k)}$-*PLANARITY.

Proof. Obviously, if one of G' and $G - Y$ is not $\mathcal{H}^{(k)}$-planar, then neither is G by heredity of the $\mathcal{H}^{(k)}$-planarity. Let us suppose that both G' and $G - Y$ are $\mathcal{H}^{(k)}$-planar. We want to prove that G is $\mathcal{H}^{(k)}$-planar. For this, we find a well-linked rendition δ of (G', Ω) and ground-maximal sphere decompositions δ' of G' and δ_Y of $G - Y$.

Let C_i be the i-th layer of W for $i \in [(r-1)/2]$ (so C_1 is the perimeter of W). Let Ω be the cyclic ordering of the vertices of $A \cap B$ as they appear in C_1. Hence, ρ is a rendition of $(G' = G[B], \Omega)$. Then, by Proposition 10.2.10, (G', Ω) has a well-linked rendition δ.

By Corollary 10.2.9, given that $r \geq \max\{\sqrt{(k+7)/2} + 2, 7\}$, there are two ground-maximal $\mathcal{H}^{(k)}$-compatible sphere decompositions $\delta' = (\Gamma', \mathcal{D}')$ and $\delta_Y = (\Gamma_Y, \mathcal{D}_Y)$ of G' and $G - Y$, respectively, such that the $(r-2)$-central wall W' of W is grounded in both δ' and δ_Y.

Then, by Lemma 10.2.15, given that $r \geq q + 10$, there exists a ground-maximal sphere decomposition δ^* of G such that each cell δ^* is either a cell of δ' or a cell of δ_Y, and is thus $\mathcal{H}^{(k)}$-compatible. Thus, by Lemma 10.2.5, G is $\mathcal{H}^{(k)}$-planar. $\square$

If we take $q = 3$, then we get the following.

Corollary 10.2.22. *Let $\mathcal{H}$ be a hereditary graph class. Let $k, r \in \mathbb{N}$ with $r \geq \max\{\sqrt{(k+7)/2} + 2, 13\}$. Let G be a graph, $(W, \mathfrak{R} = (X, Y, P, C, \rho))$ be a flatness pair of G of height r, G' be the $\mathfrak{R}$-compass of W, and v be a central vertex of W. Then G is a* yes*-instance of $\mathcal{H}^{(k)}$-*PLANARITY *if and only if G' and $G - v$ are both* yes*-instances of $\mathcal{H}^{(k)}$-*PLANARITY.

10.2.6 Proof of Theorem 10.1.4

We can finally prove Theorem 10.1.4.

Theorem 10.1.4. *Let $k \in \mathbb{N}$ and let $\mathcal{H}$ be a polynomial-time decidable hereditary graph class. Then there is an algorithm that solves $\mathcal{H}^{(k)}$-*PLANARITY *in time $f(k) \cdot n(n+m)$ for some computable function f.*

Proof. We apply Theorem 10.2.2 to G, with $k' = \lceil \sqrt{k+4} \rceil + 2$ and $r = \max\{\mathsf{odd}(\sqrt{(k+7)/2} + 2), 13\}$. It runs in time $\mathcal{O}_k(n+m)$.

If G has treewidth at most $f_{10.2.2}(k') \cdot r$, then we apply Proposition 4.3.2 to G in time $\mathcal{O}_k(n)$ and solve the problem. We can do so because the graphs in $\mathcal{H}^{(k)}$ have a bounded size, so $\mathcal{H}^{(k)}$ is a finite graph class, hence trivially CMSO-definable. Therefore, by Observation 10.0.1, $\mathcal{H}^{(k)}$-PLANARITY is expressible in CMSO logic.

If G contains an apex grid of height k' as a minor, then by Lemma 10.2.3, we obtain that G has no planar $\mathcal{H}^{(k)}$-modulator and report a no-instance.

Hence, we can assume that there is a flatness pair $(W, \mathfrak{R})$ of height r in G whose $\mathfrak{R}$-compass G' has treewidth at most $f_{10.2.2}(k') \cdot r$. Let v be a central vertex of W. We apply Proposition 4.3.2 to G' in time $\mathcal{O}_k(n)$ and we recursively apply our algorithm to $G - v$. If the outcome is a no-instance for one of them, then this is also a no-instance for G. Otherwise, the outcome is a yes-instance for both. Then, by Corollary 10.2.22, we can return a yes-instance.

The running time of the algorithm is $T(n) = \mathcal{O}_k(n+m) + T(n-1) = \mathcal{O}_k(n(n+m))$. □

Note that the running time of Proposition 4.6.3, and thus Theorem 10.2.2 can be modified so that the dependence on t (resp. k) and r is explicit. Therefore, the only reason we cannot give an explicit dependence on k here is Courcelle's theorem.

10.3 Planar elimination distance

In this section, we prove Lemma 10.1.5 in Subsection 10.3.1, and Lemma 10.1.6 in Subsection 10.3.2, after having given a necessary auxiliary result in Subsection 7.4.1.

10.3.1 Finding a big leaf in $\mathcal{H}$

In this section, we prove Lemma 10.1.5.

The algorithm uses the following result.

Proposition 10.3.1 ([60]). *Given a set V of size n and $a, b \in [0, n]$, one can construct in time $2^{\mathcal{O}(\min\{a,b\}\log(a+b))} \cdot n \log n$ a family $\mathcal{F}_{a,b}$ of at most $2^{\mathcal{O}(\min\{a,b\}\log(a+b))} \cdot \log n$ subsets of V such that the following holds: for any disjoint sets $A, B \subseteq U$ with $|A| \leq a$ and $|B| \leq b$, there exists a set $R \in \mathcal{F}$ such that $A \subseteq R$ and $B \cap R = \emptyset$.*

Proof of Lemma 10.1.5. The algorithm goes as follows. We construct the family $\mathcal{F}_{a,k'}$ of Proposition 10.3.1 in time $2^{\mathcal{O}(k\log(a+k))} \cdot n \log n$. For each $U \in \mathcal{F}$, we do the following. We construct $\mathcal{C}_U := \{C \in \mathsf{cc}(G - U) \mid C \notin \mathcal{H}\}$ in time $\mathcal{O}(n^c + n + m)$ and set $Z_U := \bigcup_{C \in \mathcal{C}_U} V(C)$. If $|A_U| \leq k'$, where $A_U := N_G(Z_U)$, then let $C_U \in \mathsf{cc}(G - A_U)$ be the unique component of size at least a, if it exists. We compute, if it exists, a minimum solution S_U of Vertex Deletion to $\mathcal{H}$ for the instance $(C_U, k' - |A_U|)$ in time $k' \cdot f(k') \cdot n^c$. For each subset $Y_U \subseteq V(G) \setminus N_G[V(C_U)]$ (which has size at most $a-1$), we set $X_U := Y_U \cup A_U \cup S_U$. We check whether $\mathsf{torso}(G, X_U) \in \mathcal{G}_k$ (in time at most $2^{(a+k)^2} + n + m$) and $G - X_U \in \mathcal{H}$ (in time time $\mathcal{O}(n^c)$). If that is the case, we return the set X_U. If, for every $U \in \mathcal{F}$, we did not return anything, then we return a no-instance.

Running time. Given that $|\mathcal{F}_{a,k'}| \leq 2^{\mathcal{O}(k\log(a+k))} \cdot \log n$, the algorithm takes time $f(k) \cdot 2^{\mathcal{O}((a+k)^2)} \cdot \log n \cdot (n^c + n + m)$.

Correctness. Obviously, if the algorithm outputs a set X_U, then it is a $\mathcal{G}_k \triangleright \mathcal{H}$-modulator of G. It remains to show that if G admits a big-leaf $\mathcal{G}_k \triangleright \mathcal{H}$-modulator, then the algorithm indeed output some set X_U. Assume that G admits a big-leaf $\mathcal{G}_k \triangleright \mathcal{H}$-modulator X with big leaf D.

We set $L := N_G[V(D)]$, $R := V(G) \setminus V(D)$, $A := L \cap R = N_G(V(D))$, and $B := R \setminus L = V(G) - N_G[V(D)]$. By Observation 10.1.2, we have $|N_G(V(D))| \leq k'$. Therefore, (L, R) is a separation of G of order at most k'. Moreover, G is (a, k')-unbreakable, so given that $|L \setminus R| = |V(D)| \geq a$, we conclude that $|B| \leq a - 1$. Therefore, by Proposition 10.3.1, we can construct in time $2^{\mathcal{O}(k\log(a+k))} \cdot n \log n$ a family $\mathcal{F}$ of at most $2^{\mathcal{O}(k\log(a+k))} \cdot \log n$ subsets of $V(G)$ such that there exists $U \in \mathcal{F}$ with $A \subseteq U$ and $B \cap U = \emptyset$.

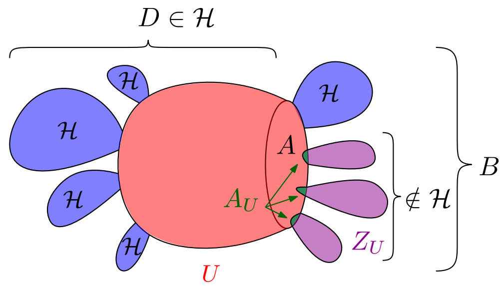

Figure 10.8: Illustration for the correctness of Lemma 10.1.5. C_U is the union of the red and the blue part.

By heredity of $\mathcal{H}$, for each $C \in \mathcal{C}_U$, C is not a subgraph of D. Therefore, $V(C) \subseteq B$, and thus $N_G(V(C)) \subseteq A$. Hence, $Z_U \subseteq B$ and $A_U \subseteq A$. In particular, $|Z_U| \leq |B| < a$. Moreover, $V(D) \subseteq V(G) \setminus A_U$, so, given that G is (a,k')-unbreakable, the component $C_U \subseteq \mathsf{cc}(G - A_U)$ such that $V(D) \subseteq V(C_U)$ is the unique component of size at least a, and $|V(G) \setminus N_G[V(C_U)]| < a$. Given that, for each $C \in \mathsf{cc}(G-U) \setminus \mathcal{C}_U$, $C \in \mathcal{H}$, and that $\mathcal{H}$ is closed under disjoint union, we conclude that $A \setminus A_U$ is a solution of Vertex Deletion to $\mathcal{H}$ for the instance $(C_U, k' - |A_U|)$. Therefore, the algorithm should find a minimum solution S_U of Vertex Deletion to $\mathcal{H}$ for the instance $(C_U, k' - |A_U|)$.

We set $Y_U := X \setminus N_G[V(C_U)]$. It remains to prove that $X_U := Y_U \cup A_U \cup S_U$ is a $\mathcal{G}_k \triangleright \mathcal{H}$-modulator of G. For each $C \in \mathsf{cc}(G - X_U)$, we have either $C \in \mathsf{cc}(G - X)$, or $V(C) \subseteq V(C_U)$. In the first case, it immediately implies that $C \in \mathcal{H}$. In the second case, it implies $N_G(V(C)) \subseteq A_U \cup S_U$. Hence, given that $C_U \in \mathsf{cc}(G - A_U)$ and that S_U is a solution of Vertex Deletion to $\mathcal{H}$ for the instance $(C_U, k' - |A_U|)$, we also conclude that $C_U \in \mathcal{H}$. It remains to prove that $\mathsf{torso}(G, X_U) \in \mathcal{G}_k$. We only write the proof for planar treedepth and planar treewidth, as the proof is simpler for treedepth and treewidth.

Claim 10.3.2. *If $\mathcal{G}_k$ is the class of graphs of planar treedepth at most k, then $\mathsf{torso}(G, X_U) \in \mathcal{G}_k$.*

Proof of claim. Given that X is a $\mathcal{G}_k \triangleright \mathcal{H}$-modulator of G, there is a certifying elimination sequence $X_1, \ldots, X_k$. We need to prove that $\mathsf{torso}(G, X_U) \in \mathcal{G}_k$, or, equivalently, that there is a certifying elimination sequence $X'_1, \ldots, X'_k$ whose union is X_U. Remember that $A' := A \setminus A_U$ is a solution of Vertex Deletion to $\mathcal{H}$ for the instance $(C_U, k' - |A_U|)$. By minimality of S_U, we have $|S_U| \leq |A'|$, so there is an injective function τ from S_U to A'. We define the partition $(X'_1, \ldots, X'_k)$ of X_U such that, for $i \in [k]$, $X'_i = X_i \setminus A' \cup \tau^{-1}(A' \cap X_i)$ (remember that $X_i \setminus A' \subseteq Y \cup A_U$ and that $\tau^{-1}(A' \cap X_i) \subseteq S_U$). We define $G'_1 = G$ and $G'_{i+1} = G'_i - X'_i$ for $i \in [k]$. It remains to prove that $\mathsf{torso}(G'_i, X'_i)$ is planar, for $i \in [k]$. Given that $|A \cap X_i| \leq 4$, we conclude that $(A_U \cup S_U) \cap X'_i$ induces at most a K_4 in $\mathsf{torso}(G'_i, X'_i)$. Moreover, $N_G(A') \subseteq A_U$, and thus $N_{\mathsf{torso}(G'_i, X'_i)}(S_U \cap X'_i) \subseteq A_U \cap X'_i$, so $\mathsf{torso}(G'_i, X'_i)$ is indeed planar. Therefore, $X'_1, \ldots, X'_k$ is indeed a certifying elimination sequence. $\diamond$

Claim 10.3.3. *If $\mathcal{G}_k$ is the class of graphs of planar treewidth at most k, then $\mathsf{torso}(G, X_U) \in \mathcal{G}_k$.*

Proof of claim. Let (T, β) be a tree decomposition of $\mathsf{torso}(G, X)$ of planar width at most k. Given that $A = N_G(V(D))$ induces a clique in $\mathsf{torso}(G, X)$, there is $t \in V(T)$ such that $A \subseteq \beta(t)$. Let $A' = A \setminus A_U$. Let (T, β') be the tree decomposition of $\mathsf{torso}(G, X_U)$ such that $\beta'(t) = \beta(t) \setminus A' \cup S_U$ and $\beta'(t') = \beta(t') \setminus A'$ for $t' \in V(T) \setminus \{t\}$. If $\beta(t)$ has size at most k, then so does $\beta'(t)$ given that

$|S_U| \leq |A'|$. If $\beta(t)$ has a planar torso, then $|A_U \cup S_U| \leq |A| \leq 4$, and $N_{\mathsf{torso}(G,X_U)}(S_U) \subseteq A_U$, so the torso of $\beta'(t)$ is also planar. Therefore, (T, β') has planar width at most k, hence the result. $\diamond$

$\square$

10.3.2 The algorithm

We can now prove Lemma 10.1.6.

Proof of Lemma 10.1.6. We set $\alpha = \sqrt{a+3}+2$, $d = \alpha^4$, $s' = a + 4k - 3$, $s = (d-1) \cdot g_{4.6.3}(k') + s'$, $z = \max\{\sqrt{(a+4k-2)/2}+6, 11\}$, $r_3 = f_{7.4.2}(4(k-1), z, 3)$, $r_2 = \lceil \sqrt{s} \cdot (r_3+1) \rceil$, and $r_1 = r_2 + 2\alpha$. The algorithm goes as follows. If $k = 0$, then it reduces to checking whether $G \in \mathcal{H}^{(a-1)}$, which can be done in time $\mathcal{O}_a(1)$ given that $\mathcal{H}^{(a-1)}$ is finite. If $k = 1$, then it reduces to checking whether G is $\mathcal{H}^{(a-1)}$-planar. Therefore, we can apply Theorem 10.1.4 in time $\mathcal{O}_a(n \cdot (n+m))$ and conclude. Hence, we now assume that $k \geq 2$. We apply the algorithm of Proposition 4.6.3 with input $(G, k'+a, r_1)$, which runs in time $\mathcal{O}_{k,a}(n)$.

If $K_{k'+a}$ is a minor of G, then we report a no-instance. We can do so because the graphs of planar treedepth at most k are K_{4k+1}-minor-free, and thus so is the torso of any $\mathcal{P}^k \triangleright \mathcal{H}^{(a-1)}$-modulator of G, and the graphs in $\mathcal{H}^{(a-1)}$ have at most $a-1$ vertices. Hence, if G has $\mathcal{H}$-planar treedepth at most k, then G is $K_{k'+a}$-minor-free.

If G has treewidth at most $f_{4.6.3}(k'+a) \cdot r_1$, then we apply Proposition 4.3.2 to G in time $\mathcal{O}_{k,a}(n)$ and solve the problem. We can do so because $\mathcal{H}^{(a-1)}$-planarity is expressible in CMSO logic, and therefore, by induction, having $\mathcal{H}^{(a-1)}$-planar treedepth at most k is also expressible in CMSO logic.

Hence, we can assume that there is a set $A \subseteq V(G)$ of size at most $g_{4.6.3}(k')$ and a flatness pair $(W_1, \mathfrak{R}_1)$ of $G - A$ of height r_1 such that $\mathsf{Compass}_{\mathfrak{R}_1}(W_1)$ has treewidth at most $f_{4.6.3}(k'+a) \cdot r_1$. Let W_2 be the central r_2-subwall of W_1.

Given that $r_2 \geq \lceil \sqrt{s} \cdot (r_3+1) \rceil$, we can find a collection $\mathcal{W}' = \{W_3'^1, \ldots, W_3'^s\}$ of r_3-subwalls of W_2 such that the sets $\mathsf{influence}_{\mathfrak{R}_i}(W_i)$ are pairwise disjoint. Then, by applying the algorithm of Proposition 4.6.6, in time $\mathcal{O}(n+m)$, we find a collection $\mathcal{W} = \{W_3^1, \ldots, W_3^s\}$ such that, for $i \in [s]$, $(W_3^i, \mathfrak{R}_3^i)$ is a $W_3'^i$-tilt of $(W_1, \mathfrak{R}_1)$, and the graphs $\mathsf{Compass}_{\mathfrak{R}_3^i}(W_3^i)$ are pairwise disjoint and have treewidth at most $f_{4.6.3}(t) \cdot r_1$.

Let A^- denote the set of vertices of A that are adjacent to vertices in the compass of at most $d-1$ walls of $\mathcal{W}$, and let $A^+ := A \setminus A^-$. A^- can be constructed in time $\mathcal{O}_k(n)$. If $|A^+| \geq 4k$, then return a no-instance, as justified later in Claim 10.3.6. Then, given that $s \geq (d-1) \cdot |A| + s'$, by the pigeonhole principle, there is $I \subseteq [s]$ of size s' such that no vertex of A^- is adjacent to the $\mathfrak{R}_3^i$-compass of W_3^i for $i \in I$. Therefore, $(W_3^i, \mathfrak{R}_3^{i\prime})$ is a flatness pair of $G - A^+$ for $i \in I$, where $\mathfrak{R}_3^{i\prime}$ is the 5-tuple obtained from $\mathfrak{R}_3^i$ by adding A^- to its first coordinate.

For $i \in I$, let F_i denote the $\mathfrak{R}_3^{i\prime}$-compass of W_3^i. Given that F_i has treewidth at most $f_{4.6.3}(t) \cdot r_1$, we apply Proposition 4.3.2 to compute in time $\mathcal{O}_{k,a}(n)$ the minimum $d_i \leq k$, if it exists, such that F_i is $(\mathcal{P}^{d_i-1} \triangleright \mathcal{H})^{(a-1)}$-planar. If, for all $i \in I$, such a d_i does not exist, then we report a no-instance as justified later in Claim 10.3.5. Otherwise, there is $p \in I$ such that d_p is minimum.

Let v be a central vertex of W_3^p. We apply recursively our algorithm to $G - v$ and return a yes-instance if and only if it returns a yes-instance.

Correctness. We now prove the correctness of the algorithm. Suppose that $G - v$ has $\mathcal{H}^{(a-1)}$-planar treedepth at most k. Let us prove that it implies that G has $\mathcal{H}^{(a-1)}$-planar treedepth at most k. Let $X_1, \ldots, X_k$ be a certifying elimination sequence of G and let $G_i := G_{i-1} - X_i$ for $i \in [k]$, where $G_0 := G - v$.

Let $j := \max\{i \in [0,k] \mid \exists C_i \in \mathsf{cc}(G_i), |V(C_i)| \geq a\}$. Given that $G_0 = G - v$ contains as a connected subgraph the graph $W_1 - v$ of size more that a, we conclude that there is $C_0 \in \mathsf{cc}(G_0)$ such that $|V(C_0)| \geq a$, so j is well-defined. Let $C_j \in \mathsf{cc}(G_j)$ be such that $|V(C_j)| \geq a$. See Figure 10.9 for an illustration. Given that, for each $C \in \mathsf{cc}(G_k)$, $C \in \mathcal{H}^{(a-1)}$, and thus $|V(C)| < a$, we conclude that $j < k$.

Claim 10.3.4. *$|V(G-v) \setminus V(C_j)| \leq a + 4k - 5$ and C_j is $(\mathcal{P}^{k-j-1} \triangleright \mathcal{H})^{(a-1)}$-planar.*

Proof of claim. Given that $X_1, \ldots, X_k$ is a certifying elimination sequence of $G - v$, it implies that C_j has $\mathcal{H}^{(a-1)}$-planar treedepth at most $k - j$ and that $X_j \cap V(C_j)$ is a planar $\mathcal{P}^{k-j-1} \triangleright \mathcal{H}^{(a-1)}$-modulator of C_j. Since G is an (a, k')-unbreakable graph and that $k' = 4k$, it implies that $G - v$ is $(a, 4k-1)$-unbreakable. Note that $(N_{G-v}[V(C_j)], V(G-v) \setminus V(C_j))$ is a separation of order at most $|N_{G-v}(V(C_j))| \leq 4j \leq 4(k-1) < 4k-1$, which implies that $|V(G-v) \setminus V(C_j)| \leq a-1+4j \leq a+4k-5$. By maximality of j, for each $C \in \mathsf{cc}(C_j - X_{j+1}) \subseteq \mathsf{cc}(G_{j+1})$, $|V(C)| < a$. Therefore, given that $X_{j+1} \cap V(C_j)$ is a planar $\mathcal{P}^{k-j-1} \triangleright \mathcal{H}^{(a-1)}$-modulator of C_j, it is also a planar $(\mathcal{P}^{k-j-1} \triangleright \mathcal{H})^{(a-1)}$-modulator of C_j. $\diamond$

The next claim justify that we can report a no-instance if, for all $i \in I$, there is no $d_i \leq k$ such that F_i is $(\mathcal{P}^{d_i-1} \triangleright \mathcal{H})^{(a-1)}$-planar.

Claim 10.3.5. *$d_p \leq k - j$.*

Proof of claim. Given that, by Claim 10.3.4, $|V(G-v) \setminus V(C_j)| \leq a + 4k - 5 \leq s' - 2$, it implies, by the pigeonhole principle, that there is $q \in I \setminus \{p\}$ such that F_q is entirely contained in C_j. Hence, by Claim 10.3.4, F_q is $(\mathcal{P}^{k-j-1} \triangleright \mathcal{H})^{(a-1)}$-planar, and therefore, $d_q \leq k - j$. Additionally, by minimality of d_p, $d_q \geq d_p$. Therefore, $d_p \leq k - j$. $\diamond$

We now prove that we have a no-instance when $|A^+| \geq 4k$.

Claim 10.3.6. *$A^+ \subseteq N_G(V(C_j))$, and hence $|A^+| \leq 4k - 1$.*

Proof of claim. Let $u \in A^+$. u is adjacent to the compass of at least $d \geq \alpha^4$ walls of $\mathcal{W}$. These compasses are connected and pairwise disjoint, and are contained in the central r_2-subwall of W_1, where $r_2 = r_1 - 2\alpha$. Then, observe that G contains as a minor an $(r_1 \times r_1)$-grid (obtained by contracting the intersection of horizontal and vertical paths of W_1) along with a vertex (corresponding to u) that is adjacent to d vertices of its central $(r_2 \times r_2)$-subgrid (corresponding to W_2). Thus, by Proposition 10.2.1, G contains a model of an apex grid Γ_α^+ of height α, where the branch set of the universal vertex is the singleton $\{u\}$. Therefore, by Lemma 10.2.3, given that $\alpha \geq \sqrt{a+3} + 2$, and that C_j admits a planar $(\mathcal{P}^{k-j-1} \triangleright \mathcal{H})^{(a-1)}$-modulator, we conclude that $u \notin V(C_j)$. Given that $|V(G-v) \setminus V(C_j)| \leq a + 4k - 5$ and that $s \geq a + 4k - 4$, u neighbors at least one vertex of $V(C_j)$, so we conclude that $u \in N_G(V(C_j))$. Therefore, $|A^+| \leq |N_G(V(C_j))| \leq |N_{G-v}(V(C_j))| + 1 \leq 4k - 1$. $\diamond$

We set $S := N_{G-v}(V(C_j))$, which has size at most $4(k-1)$. Remember that $(W_3^p, \mathfrak{R}_3^p)$ is a flatness pair of $G - A^+$ of height $r_3 \geq f_{7.4.2}(4(k-1), z, 3)$. Note also that v is contained in the compass of every $W^{(3)}$-tilt of $(W_3^p, \mathfrak{R}_3^p)$. By Lemma 7.4.2, there is a flatness pair $(W^*, \mathfrak{R}^*)$ of $G - A^+$ that is a $\tilde{W}$-tilt of $(W_3^p, \mathfrak{R}_3^p)$ for some z-subwall $\tilde{W}$ of W_3^p such that v and $S \cap V(\mathsf{Compass}_{\mathfrak{R}^*}(W^*))$ are contained in the compass of every $W^{*(5)}$-tilt of $(W^*, \mathfrak{R}^*)$. Let Y be the vertex set of the compass of some $W^{*(5)}$-tilt $(W_Y, \mathfrak{R}_Y)$ of $(W^*, \mathfrak{R}^*)$.

We set $G' := \mathsf{Compass}_{\mathfrak{R}^*}(W^*)$. Let G^* be the graph induced by $V(C_j)$ and Y, and let $\mathfrak{R}''$ be the 5-tuple obtained from $\mathfrak{R}^*$ after removing $V(G) \setminus V(C_j)$ from its first coordinate.

Claim 10.3.7. *$(W^*, \mathfrak{R}'')$ is a flatness pair of G^* and $G' = \mathsf{Compass}_{\mathfrak{R}''}(W^*)$.*

Proof of claim. Remember that Y is the vertex set of the $\mathfrak{R}_Y$-compass of the flatness pair $(W_Y, \mathfrak{R}_Y = (A_Y, Y, P_Y, C_Y, \rho_Y))$ of $G - A^+$ and that G' is the $\mathfrak{R}^*$-compass of the flatness pair $(W^*, \mathfrak{R}^* = (A^*, V(G'), P^*, C^*, \rho^*))$ of $G - A^+$. Hence, $(W^*, \mathfrak{R}'' = (A^* \cap V(C_j), V(G'), P^*, C^*, \rho^*))$ is a flatness pair of $G - A^+ - (A^* \setminus V(C_j))$. Let us prove that $G - A^+ - (A^* \setminus V(C_j)) = G^*$. Given that $V(G^*) = V(C_j) \cup Y$ and that $Y \subseteq V(G') \setminus A^*$, we trivially have that G^* is an induced subgraph of $G - A^+ - (A^* \setminus V(C_j))$.

For the other direction, it is enough to prove that $V(G') \subseteq V(C_j) \cup Y$, given that $A^* \cap V(C_j) \subseteq V(C_j)$. Given that W^* has height z and that W_Y has height five, $W^* - V(W_Y)$, and thus $G' - Y$, contains a wall of height $z - 6$ has a subgraph. Thus, $|V(G') \setminus Y| \geq 2(z-6)^2 - 2 \geq a + 4k - 4$. Hence, by Claim 10.3.4, we conclude that $|(V(G') \setminus Y) \cap V(C_j)| \geq 1$. Remember that $(N_{G-v}[V(C_j)], V(G - v) \setminus V(C_j))$ is a separation of $G - v$ with separator $N_{G-v}(V(C_j)) = S$. Given that $S \cap V(G') \subseteq Y$, it thus implies that $V(G') \cap (V(G - v) \setminus V(C_j)) \subseteq Y$ by connectivity of the compass. Since we also have $v \in Y$, we conclude that $V(G') \subseteq V(C_j) \cup Y$, and thus that $(W^*, \mathfrak{R}'')$ is a flatness pair of G^*. ◇

Remember that $v \in Y$. Let us show that G^* has $\mathcal{H}^{(a-1)}$-planar treedepth at most $k - j - 1$. We will later combine this result with the fact that $G - v$ has $\mathcal{H}^{(a-1)}$-planar treedepth at most k to prove that G has $\mathcal{H}^{(a-1)}$-planar treedepth at most k.

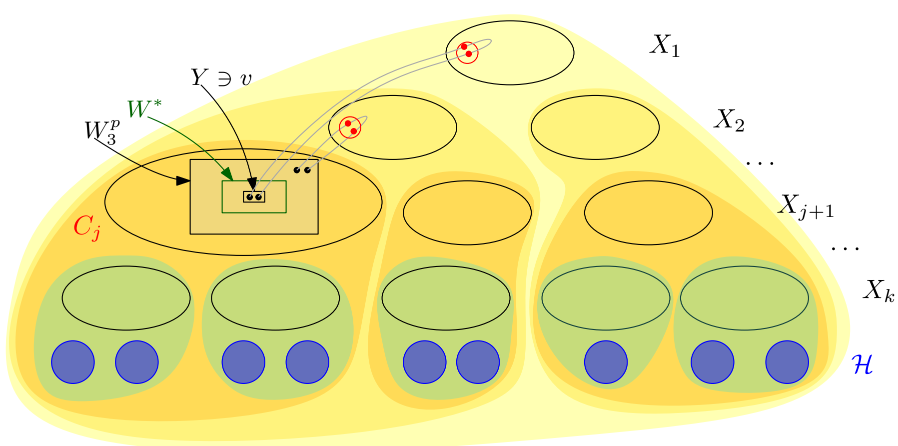

Figure 10.9: Illustration for the correctness of Lemma 10.1.6.

Claim 10.3.8. *G^* has $\mathcal{H}^{(a-1)}$-planar treedepth at most $k - j$.*

Proof of claim. As discussed previously, C_j is $(\mathcal{P}^{(k-j-1)} \triangleright \mathcal{H})^{(a-1)}$-planar. Given that $G^* - Y$ is an induced subgraph of C_j, it is also $(\mathcal{P}^{(k-j-1)} \triangleright \mathcal{H})^{(a-1)}$-planar by heredity. By Claim 10.3.7, $(W^*, \mathfrak{R}'')$ is a flatness pair of G^* and $G' = \mathsf{Compass}_{\mathfrak{R}''}(W^*)$. Given that $(W^*, \mathfrak{R}^*)$ is a $\tilde{W}$-tilt of $(W_3^p, \mathfrak{R}_3^p)$, G' is an induced subgraph of F_p, and thus, by Claim 10.3.5, G' is $(\mathcal{P}^{(k-j-1)} \triangleright \mathcal{H})^{(a-1)}$-planar. Therefore, by Corollary 10.2.21 applied for the graph G^*, G', and Y, given that $z \geq \max\{\sqrt{(a+6)/2} + 2, 11\}$, we conclude that G^* is $(\mathcal{P}^{(k-j-1)} \triangleright \mathcal{H})^{(a-1)}$-planar. In particular, it means that G^* has $\mathcal{H}^{(a-1)}$-planar treedepth at most $k - j$. ◇

Claim 10.3.9. *G has $\mathcal{H}^{(a-1)}$-planar treedepth at most k.*

Proof of claim. Given that G^* has $\mathcal{H}^{(a-1)}$-planar treedepth at most $k - j$, there exists a certifying elimination sequence $Y_1, \dots, Y_{k-j}$. We define $(X_1^*, \dots, X_k^*)$ and $(G_0^*, G_1^*, \dots, G_k^*)$ as follows. For $i \in [j]$, $X_i^* := X_i \setminus Y$. For $i \in [k-j]$, we set $X_{j+i}^* := X_{j+i} \setminus V(C_j) \cup Y_i$. Finally, we set $G_0^* := G$ and

$G_i^* := G_{i-1}^* - X_i^*$ for $i \in [k]$. We want to prove that prove that $X_1^*, \ldots, X_k^*$ is a certifying elimination sequence of G, i.e. that $\mathsf{torso}(G_{i-1}^*, X_i^*)$ is planar for $i \in [k]$ and that $C \in \mathcal{H}^{(a-1)}$ for $C \in \mathsf{cc}(G_k^*)$.

For each $D \in \mathsf{cc}(G_j^*)$, either $D = G^*$, or $D \in \mathsf{cc}(G_j - Y)$. Therefore, for each $C \in \mathsf{cc}(G_k^*)$, either (a) $V(C) \subseteq V(G^*)$, in which case $C \in \mathsf{cc}(G^* - \bigcup_{i\in[k-j]} Y_i)$ and thus $C \in \mathcal{H}^{(a-1)}$, given that $Y_1, \ldots, Y_{k-j}$ is a certifying elimination sequence of G^*, or (b) there is $D \in \mathsf{cc}(G_j - Y)$ such that $V(C) \subseteq V(D)$, in which case $C \in \mathsf{cc}(G_k - Y)$ and thus again $C \in \mathcal{H}^{(a-1)}$ by heredity of $\mathcal{H}^{(a-1)}$.

Given that $(W^*, \mathfrak{R}^*)$ is a flatness pair of $G - A^+$ and that Y is the vertex set of some $W^{*(5)}$-tilt of $(W^*, \mathfrak{R}^*)$, it implies that $N_G(Y) \subseteq V(G') \cup A^+$. Additionally, by Claim 10.3.7, $V(G') \subseteq V(C_j) \cup Y$, and, by Claim 10.3.6, $A^+ \subseteq N_G(V(C_j))$, so $N_G(Y) \subseteq N_G[V(C_j)]$. So finally, given that $v \in Y$, we conclude that $N_G(Y) \subseteq N_{G-v}[V(C_j)]$.

For $i \in [j]$, $X_i^* = X_i \setminus Y$. $N_G(Y) \cap X_i$ already induces a clique in $\mathsf{torso}(G_{i-1}, X_i)$ because $N_{G-v}[V(C_j)] \cap X_i = N_{G-v}(V(C_j)) \cap X_i$ induces a clique in $\mathsf{torso}(G_{i-1}, X_i)$. Therefore, $\mathsf{torso}(G_{i-1}^*, X_i^*)$ is a subgraph of $\mathsf{torso}(G_{i-1}, X_i)$, that is thus planar. For $i \in [k-j]$, the connected components of X_{j+i}^* are either connected components of Y_i or connected components of X_{j+i}, so their torso is planar in either case. Hence the result. $\diamond$

$\square$

10.4 $\mathcal{H}$-planar treewidth

In Subsection 10.4.1, we prove that a graph has $\mathcal{H}$-planar treewidth at most k if and only if it has ground-maximal sphere decomposition whose cells have property $\Pi_{\mathcal{H},k}$. From this, we deduce in Subsection 10.4.2 a proof of Lemma 10.1.7.

10.4.1 Expression as a sphere decomposition

Quasi-4-connectivity. Given $k \in \mathbb{N}_{\geq 1}$, a graph G is k-connected if for all separation (L, R) of order at most $k-1$ of G, either $L \subseteq R$ or $R \subseteq L$. A graph G is *quasi-4-connected* if it is 3-connected and that for all separation (L, R) of order three of G, either $|L \setminus R| \leq 1$ or $|R \setminus L| \leq 1$.

Proposition 10.4.1 ([150]). *Every G has a tree decomposition (T, β) of adhesion at most three such that, for each $t \in V(T)$, $\mathsf{torso}(G, \beta(t))$ is a minor of G that is quasi-4-connected.*

Lemma 10.4.2. *Let $\mathcal{H}$ be a graph class and $k \in \mathbb{N}$. Let G be a graph of $\mathcal{H}$-planar treewidth at most k and X be a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator of G. Then $\mathsf{torso}(G, X)$ has a tree decomposition (T, β) of adhesion at most three such that for each $t \in V(T)$, $\mathsf{torso}(G, \beta(t))$ is a minor of $\mathsf{torso}(G, X)$ that is quasi-4-connected, and either it is planar or it has treewidth at most k.*

Proof. By Proposition 10.4.1, there is a tree decomposition (T, β) of $\mathsf{torso}(G, X)$ of adhesion at most three such that, for each $t \in V(T)$, $\mathsf{torso}(G, \beta(t))$ is a minor of $\mathsf{torso}(G, X)$ that is quasi-4-connected. Let $t \in V(T)$. Given that $\mathsf{torso}(G, X)$ has planar treewidth at most k and that $\mathsf{torso}(G, \beta(t))$ is a minor of $\mathsf{torso}(G, X)$, it implies that $\mathsf{torso}(G, \beta(t))$ has planar treewidth at most k. Thus, there is a tree decomposition (T^t, β^t) of $\mathsf{torso}(G, \beta(t))$ such that, for each $u \in V(T^t)$, $\mathsf{torso}(G, \beta^t(u))$ either is planar or has treewidth at most k. We chose (T^t, β^t) to have the minimum number of nodes. This implies in particular that, for any $u, u' \in V(T^t)$, $\beta^t(u) \setminus \beta^t(u') \neq \emptyset$.

Suppose towards a contradiction that there are two adjacent nodes u, u' in $V(T^t)$ and that one of them, say u, is such that $\mathsf{torso}(G, \beta^t(u))$ is planar. Then $|\mathsf{adh}(u, u')| \leq 4$ and, by planarity of $\mathsf{torso}(G, \beta^t(u))$, there is $Y \subseteq \mathsf{adh}(u, u')$ that is a 3-separator in $\mathsf{torso}(G, \beta(t))$. Then, by quasi-4-connectivity of $\mathsf{torso}(G, \beta(t))$, $|Y| = |\mathsf{adh}(u, u')| = 3$ and either $|\beta^t(u) \setminus Y| = 1$ or $|\beta^t(u') \setminus Y| = 1$.

If $\beta^t(u') \setminus Y = \{v\}$, then, again by quasi-4-connectivity of $\mathsf{torso}(G, \beta(t))$, Y induces a face of $\mathsf{torso}(G, \beta^t(u))$, so $\mathsf{torso}(G, \beta^t(u) \cup \{v\})$ is also planar, given that $N_{\beta(t)}(v) = Y$. Therefore, the tree decomposition obtained by removing u' and adding v to $\beta^t(u)$ is a tree decomposition of $\mathsf{torso}(G, \beta(t))$ such that, for each $u \in V(T^t)$, $\mathsf{torso}(G, \beta^t(u))$ either is planar or has treewidth at most k, and with less nodes as (T^t, β^t), a contradiction.

Assume now that $\beta^t(u) \setminus Y = \{v\}$ and $|\beta^t(u') \setminus Y| > 1$. By symmetry, if $\mathsf{torso}(G, \beta^t(u'))$ is planar, we get a contradiction, so we assume that $\mathsf{torso}(G, \beta^t(u'))$ has treewidth at most k. But then, $\mathsf{torso}(G, \beta^t(u') \cup \{v\})$ also has treewidth at most k, so we can again construct a tree decomposition of $\mathsf{torso}(G, \beta(t))$ such that, for each $u \in V(T^t)$, $\mathsf{torso}(G, \beta^t(u))$ either is planar or has treewidth at most k, and with less nodes as (T^t, β^t), a contradiction.

Therefore, for each $t \in V(T_1)$, either $\mathsf{torso}(G, \beta(t))$ is planar, or $\mathsf{torso}(G, \beta(t))$ has treewidth at most k. Hence the result. □

Property $\Pi_{\mathcal{H},k}$. Let $\mathcal{H}$ be a graph class and $k \in \mathbb{N}$. Given a sphere decomposition δ of a graph G, we say that a cell $c \in C(\delta)$ has the *property* $\Pi_{\mathcal{H},k}$ if G_c has a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator X_c such that $\pi_\delta(\tilde{c}) \subseteq X_c$, where G_c is there graph obtained from $\sigma(c)$ by making a clique out of $\pi_\delta(\tilde{c})$. We say that δ has the *property* $\Pi_{\mathcal{H},k}$ if each cell of δ has the property $\Pi_{\mathcal{H},k}$.

Lemma 10.4.3. *Let $\mathcal{H}$ be a graph class and $k \in \mathbb{N}$. Let G be a graph. Suppose that G has a sphere decomposition δ with the property $\Pi_{\mathcal{H},k}$. Then G has $\mathcal{H}$-planar treewidth at most k.*

Proof. We claim that $X := \bigcup_{c \in C(\delta)} X_c$ is a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator of G. Indeed, for each $c \in C(\delta)$, let (T_c, β_c) be a tree decomposition of $\mathsf{torso}(G_c, X_c)$ of planar width at most k. Since $\pi_\delta(\tilde{c}) \subseteq X_c$ induces a clique in G_c, there is $t_c \in V(T_c)$ such that $\pi_\delta(\tilde{c}) \subseteq \beta_c(t_c)$. We define a tree decomposition of $\mathsf{torso}(G, X)$ as follows. We set T to be the union of the trees T_c for $c \in C(\delta)$ and a new vertex t with an edge tt_c for each $c \in C(\delta)$. Obviously, T is a tree. We set β to be the function such that $\beta(t) = \pi_\delta(N(\delta))$ and, for $c \in C(\delta)$, $\beta|_{V(T_c)} = \beta_c$. (T, β) is a tree decomposition of $\mathsf{torso}(G, X)$ such that $\mathsf{torso}(G, \beta(t))$ is planar, by definition of a sphere decomposition. Hence the result. □

Lemma 10.4.4. *Let $\mathcal{H}$ be a hereditary graph class and let $a, k, r \in \mathbb{N}$ with $r \geq \max\{a + 3, k + 1, 7\}$. Let G be an $(a, 3)$-unbreakable graph and W be an r-wall of G. Suppose that G has $\mathcal{H}^{(a-1)}$-planar treewidth at most k. Then G has a sphere decomposition δ with the property $\Pi_{\mathcal{H},k}$ such that the $(r-2)$-central wall W' of W is grounded in δ and such that each cell of δ has size at most $a - 1$.*

Proof. Let X be a $\mathcal{PT}_k \triangleright \mathcal{H}^{(a-1)}$-modulator of G. Let $V' \subseteq V(W')$ be the set of 3-branch vertices of W that are vertices of W'.

By Lemma 10.4.2, $\mathsf{torso}(G, X)$ has a tree decomposition (T, β) of adhesion at most three such that for each $t \in V(T)$, either $\mathsf{torso}(G, \beta(t))$ is planar and quasi-4-connected, or $\mathsf{tw}(\mathsf{torso}(G, \beta(t))) \leq k$. By Claim 10.2.6, for each $tt' \in E(T)$, there is one connected component $D \in \mathsf{cc}(G - \mathsf{adh}(t, t'))$ that contains all but at most one vertex of $V' \setminus \mathsf{adh}(t, t')$, and $G - V(D)$ contains no cycle of W'. Given that G is $(a, 3)$-unbreakable, that $|\mathsf{adh}(t, t')| \leq 3$ and that $|V(D)| \geq |V' \setminus \mathsf{adh}(t, t')| \geq a$, we conclude that $|V(G) \setminus V(D)| < a$. Therefore, there is $t \in V(T)$ such that, for each $t' \in V(T)$ adjacent to t, the connected component $D_{t'}$ containing $\beta(t) \setminus \mathsf{adh}(t, t')$ is such that $G - V(D_{t'})$ contains no cycle of W' and $|V(G) \setminus V(D_{t'})| < a$.

Claim 10.4.5. $\mathsf{torso}(G, \beta(t))$ *is planar.*

Proof of claim. Suppose towards a contradiction that $\mathsf{tw}(\mathsf{torso}(G, \beta(t))) \leq k$ and let (T_0, β_0) be a tree decomposition of $\mathsf{torso}(G, \beta(t))$ of width at most k. For each $tt' \in E(T)$, there is $u_{t'} \in V(T_0)$ such that $\mathsf{adh}(t, t') \subseteq \beta_0(u_{t'})$. Let $\mathcal{F}$ be the set of $H \in \mathsf{cc}(G - X)$ such that there is no $tt' \in E(T)$ such that $V(H) \subseteq V(D_{t'})$. Again, for each $H \in \mathcal{F}$, there is $u_H \in V(T_0)$ such that $N_G(V(H)) \subseteq \beta(u_H)$. Hence, we can define the tree decomposition (T', β') of G where T' is obtained from T_0 by adding a vertex $v_{t'}$ adjacent to $u_{t'}$ for each $tt' \in E(T)$ and a vertex v_H adjacent to u_H for each $H \in \mathcal{F}$, and β' defined by $\beta'|_{V(T_0)} = \beta_0$, $\beta'(v_{t'}) = V(D_{t'}) \cup \mathsf{adh}(t, t')$ for $tt' \in E(T)$, and $\beta'(v_H) = N_G[V(H)]$ for $H \in \mathcal{F}$. Given that $|\beta'(v_{t'})| \leq a + 2$ for $tt' \in E(T)$, that $|\beta'(v_H)| \leq a + 3$ for $H \in \mathcal{F}$, and that $|\beta'(u)| \leq k + 1$ for $u \in V(T_0)$, it implies that G has treewidth at most $\max\{a + 2, k\}$. This contradicts the fact that $\mathsf{tw}(G) \geq \mathsf{tw}(W) = r \geq 1 + \max\{a + 2, k\}$. $\diamond$

We set $S := \beta(t)$. Let $\delta = (\Gamma, \mathcal{D})$ be a sphere embedding of $\mathsf{torso}(G, S)$. Remember that (T, β) has adhesion at most three, so, for $D \in \mathsf{cc}(G - S)$, $|N_G(V(D))| \leq 3$. As in Claim 10.2.7, for each $D \in \mathsf{cc}(G - S)$, there is a δ-aligned disk Δ_D such that:

- $N_G(V(D)) = \pi_\delta(N(\delta) \cap \Delta_D)$ and
- the graph induced by $B_D := V(\mathsf{inner}_\delta(\Delta_D) \cap V(Z_D))$ contains no cycle of W'.

Note that to prove this, we use the fact that $r - 2 \geq 5$ and $|V'| \geq 2(r - 2)^2 - 2 \geq a + 4$. Also, if $N_G(V(D)) = N_G(V(D'))$, then we can assume that $\Delta_D = \Delta_{D'}$.

Let $\mathcal{D}^*$ be the inclusion-wise maximal elements of $\mathcal{D} \cup \{\Delta_D \mid D \in \mathsf{cc}(G - S)\}$. By maximality of $\mathcal{D}^*$ and planarity of $\mathsf{torso}(G, S)$, any two distinct $\Delta_D, \Delta_{D'} \in \mathcal{D}^*$ may only intersect on their boundary. For each $C \in \mathsf{cc}(G - S)$, we draw C in a $\Delta_D \in \mathcal{D}$ such that $V(C) \subseteq B_D$, and add the appropriate edges with $\pi_\delta(N(\delta) \cap \mathsf{bd}(\Delta_D))$. We similarly draw the edges of $G[S]$ to obtain a drawing Γ^* of G.

For each cell c of $\delta^* = (\Gamma^*, \mathcal{D}^*)$, there is $D \in \mathsf{cc}(G - S)$ such that $\sigma_{\delta^*}(c)$ contains no cycle of W', since $\sigma_{\delta^*}(c)$ is a subgraph of $G[B_D]$, so W' is grounded in δ^*.

It remains to prove that δ^* has property $\Pi_{\mathcal{H},k}$. Let $c \in C(\delta^*)$ and $X_c := X \cap V(\sigma_{\delta^*}(c))$. Let (T_c, β_c) be the restriction of (T, β) to X_c. Either $\sigma_{\delta^*}(c)$ is an edge of G, or there is $D \in \mathsf{cc}(G - S)$ such that $\pi_{\delta^*}(\tilde{c}) = N_G(V(D))$. Given that (T, β) is a tree decomposition of $\mathsf{torso}(G, X)$ and that $N_G(V(D))$ is a clique in $\mathsf{torso}(G, X)$ for $D \in \mathsf{cc}(G - S)$, we conclude that (T_c, β_c) is a tree decomposition of G_c where G_c is the graph obtained from $\sigma_{\delta^*}(c)$ by making a clique out of $\pi_{\delta^*}(\tilde{c})$. Additionally, given that $S \subseteq X$, we have $\pi_{\delta^*}(\tilde{c}) \subseteq X_c$. Finally, by heredity, (T_c, β_c) is a tree decomposition of $\mathcal{H}$-planar treewidth at most k. Hence the result. $\square$

Lemma 10.4.6. *Let $\mathcal{H}$ be a hereditary graph class and $k \in \mathbb{N}$. Let G be a graph. If G has a sphere decomposition δ with the property $\Pi_{\mathcal{H},k}$ that is not ground-maximal, then G has sphere decomposition δ' with the property $\Pi_{\mathcal{H},k}$ that is strictly more grounded than δ.*

Proof. Let $\delta = (\Gamma, \mathcal{D})$ be a sphere decomposition of G with the property $\Pi_{\mathcal{H},k}$ that is not ground-maximal. Hence, there is a cell $c \in C(\delta)$ that is not ground-maximal. Since c has the property $\Pi_{\mathcal{H},k}$, there is a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator X_c of $\mathsf{torso}(G_c, \sigma(c))$ such that $\pi_\delta(\tilde{c}) \subseteq X_c$. By Lemma 10.4.2, there is a tree decomposition (T, β) of $\mathsf{torso}(G_c, X_c)$ of adhesion at most three such that, for each $t \in V(T)$, $\mathsf{torso}(G_c, \beta(t))$ is a minor of $\sigma(c)$ that is quasi-4-connected, and either it is planar or it has treewidth at most k. Without loss of generality, we can assume that, for each $t, t' \in V(T)$, $\beta(t) \setminus \beta(t') \neq \emptyset$. Given that $\pi_\delta(\tilde{c})$ is a clique in $\mathsf{torso}(G_c, \sigma(c))$, there is $r \in V(T)$ such that $\pi_\delta(\tilde{c}) \subseteq \beta(r)$. We root T at r.

Suppose that there is a child t of r such that $\mathsf{adh}(r, t) \subsetneq \pi_\delta(\tilde{c})$. Let $\mathcal{T}$ be the set of all children of r such that $\mathsf{adh}(r, t') = \mathsf{adh}(r, t)$. Let V' be the set of all vertices of $G - \beta(r)$ that belong to a bag of a subtree of T rooted at a node in $\mathcal{T}$. Let δ' be the sphere decomposition of G obtained by removing c from δ and instead adding two cells c_1 and c_2 such that $\pi_{\delta'}(\tilde{c}_1) = \mathsf{adh}(t, r)$, $\sigma_{\delta'}(c_1) = G[V' \cup \mathsf{adh}(r, t')]$, $\pi_{\delta'}(\tilde{c}_2) = \pi_\delta(\tilde{c})$, and $\sigma_{\delta'}(c_1) = \sigma_\delta(c) - V'$. Note that, for $i \in [2]$,

$X_{c_i} := X_c \cap \sigma_{\delta'}(c_i)$ is a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator of $\mathsf{torso}(G_c, \sigma_{\delta'}(c_i))$. Thus, δ' is more grounded than δ and has the property $\Pi_{\mathcal{H},k}$.

Suppose that $\beta(r) = \pi_\delta(\tilde{c})$. If there is a child t of r, then we have $\mathsf{adh}(r,t) \subseteq \beta(r) = \pi_\delta(\tilde{c})$, and so $\mathsf{adh}(r,t) = \beta(r)$ given that we already handled the case when $\mathsf{adh}(r,t) \subsetneq \pi_\delta(\tilde{c})$. Therefore, $\beta(r) \subseteq \beta(t)$, which contradicts the fact that $\beta(r) \setminus \beta(t) \neq \emptyset$. Thus, r is the only node of T. Then c is $\mathcal{H}$-compatible, so by Lemma 10.2.8, for any sphere decomposition δ' that is more grounded than δ, the cells of δ' contained in c are $\mathcal{H}$-compatible and thus have property $\Pi_{\mathcal{H},k}$. So we can assume that $\beta(r) \setminus \pi_\delta(\tilde{c}) \neq \emptyset$.

Suppose that $|\tilde{c}| \leq 2$. Let $v \in \beta(r) \setminus \pi_\delta(\tilde{c})$ be a vertex adjacent to a vertex of $\pi_\delta(\tilde{c})$ in $\mathsf{torso}(G_c, \beta(r))$. It exists given that $\mathsf{torso}(G_c, \beta(r))$ is quasi-4-connected, and thus connected. Let δ' be the the sphere decomposition of G obtained by adding a new point u in the boundary $\tilde{c}$ of c, and setting $\pi_{\delta'}(u) = v$. We still have $\pi_{\delta'}(\tilde{c}) \subseteq X_c$, and δ' is a sphere decomposition of G with property $\Pi_{\mathcal{H},k}$ that is more grounded than δ.

We can thus suppose that $|\tilde{c}| = 3$. Note that, if there is a child t of r such that $\pi_\delta(\tilde{c}) = \mathsf{adh}(r,t)$, then, given that $\beta(r) \setminus \mathsf{adh}(r,t) \neq \emptyset$ and $\beta(t) \setminus \mathsf{adh}(r,t) \neq \emptyset$, it implies that c is already ground-maximal, so we can assume that $\pi_\delta(\tilde{c}) \neq \mathsf{adh}(r,t)$ for all $t \in V(T)$. If $\mathsf{torso}(G_c, \beta(r))$ is planar, given that is is also quasi-4-connected, $\pi_\delta(\tilde{c})$ either induces a face of $\mathsf{torso}(G_c, \beta(r))$, or there is a vertex v such that $(\pi_\delta(\tilde{c}) \cup \{v\}, V(\mathsf{torso}(G_c, \beta(r))) \setminus \{v\})$ is a separation of $\mathsf{torso}(G_c, \beta(r))$. In the second case, then again, c is already ground-maximal. In the first case, then let δ_c be a sphere embedding of $\mathsf{torso}(G_c, \beta(r))$ whose outer face has vertex set $\pi_\delta(\tilde{c})$. Then the sphere decomposition obtained by replacing c with δ_c still has property $\Pi_{\mathcal{H},k}$, and it is more grounded than δ.

We now suppose that $\mathsf{torso}(G_c, \beta(r))$ has treewidth at most k. Let δ' be a sphere decomposition of G more grounded than δ such that each cell $c' \in C(\delta) \setminus \{c\}$ is equivalent to a cell of $C(\delta')$. We choose δ' such that the number of ground vertices $|N(\delta')|$ is minimum among all sphere decompositions of G that are more grounded than δ and distinct from δ. Suppose towards a contradiction that there is a cell $c' \in C(\delta')$ contained in c and $v \in \pi_{\delta'}(\tilde{c}')$ such that $v \notin \beta(r)$. Let $t \in V(T)$ be the child of r whose subtree contains a node t' with $v \in \beta(t')$. Thus, $\mathsf{adh}(r,t)$ is a separator of size at most three between v and $\pi_\delta(\tilde{c})$.

If $\mathsf{adh}(r,t) \subseteq \pi_{\delta'}(N(\delta'))$, then there is δ'-aligned disk Δ whose boundary is $\mathsf{adh}(r,t)$. Let δ'' be the sphere decomposition of G obtained by removing the cells of δ' in Δ and adding instead a unique cell with boundary $\mathsf{adh}(r,t)$. δ'' is more grounded than δ and, given that $\mathsf{adh}(r,t) \neq \pi_\delta(\tilde{c})$, δ'' is distinct from δ, a contradiction to the minimality of δ'.

Otherwise, there is $x \in \mathsf{adh}(r,t) \setminus \pi_{\delta'}(N(\delta'))$. Given that $\mathsf{torso}(G, \beta(r))$ is quasi-4-connected, there are three internally vertex-disjoint paths in $\mathsf{torso}(G_c, \beta(r))$ from x to the three vertices in $\pi_\delta(\tilde{c})$. Additionally, there is a path from x to v disjoint from $\beta(r)$ (aside from its endpoint x). Thus, given that $\mathsf{torso}(G, \beta(r))$ is a minor of $\sigma(c)$, that are four internally vertex-disjoint paths in $\sigma(c)$ from x to v and the three vertices in $\pi_\delta(\tilde{c})$, respectively. This implies that $x \in \sigma_{\delta'}(c'')$ for some cell c'' with $|\tilde{c}''| \geq 4$, a contradiction to the fact that $|\tilde{c}''| \leq 3$.

Therefore, for any $c' \in C(\delta')$, $\pi_{\delta'}(\tilde{c}') \subseteq \beta(r)$. Given that $\mathsf{torso}(G_c, \beta(r))$ has treewidth at most k, by heredity of the treewidth, we easily get that $X_{c'} := X_c \cap \sigma_{\delta'}(c')$ is a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator of $\mathsf{torso}(G_c, \sigma_{\delta'}(c'))$. Hence the result. □

From the previous results of this section and Lemma 10.2.15, we deduce an irrelevant vertex technique tailored for $\mathcal{H}$-planar treewidth.

Lemma 10.4.7. *Let $\mathcal{H}$ be a hereditary graph class. Let $a, k, r \in \mathbb{N}$ with odd $r \geq \max\{a+3, k+1, 7\}$. Let G be an $(a,3)$-unbreakable graph, $(W, \mathfrak{R})$ be a flatness pair of G of height r, G' be the $\mathfrak{R}$-compass of W, and v be a central vertex of W. Then G has $\mathcal{H}^{(a-1)}$-planar treewidth at most k if and only if G' and $G - v$ both have $\mathcal{H}^{(a-1)}$-planar treewidth at most k.*

Proof. Obviously, if one of G' and $G - v$ is has $\mathcal{H}^{(a-1)}$-planar treewidth at least $k+1$, then so does G by heredity of the $\mathcal{H}^{(a-1)}$-planar treewidth. Let us suppose that both G' and $G - v$ have $\mathcal{H}^{(a-1)}$-planar treewidth at most k. We want to prove that G has $\mathcal{H}^{(a-1)}$-planar treewidth at most k. For this, we find a well-linked rendition δ of (G', Ω) and ground-maximal sphere decompositions δ' of G' and δ_v of $G - v$.

Let C_i be the i-th layer of W for $i \in [(r-1)/2]$ (so C_1 is the perimeter of W). Let $\mathcal{R} = (X, Y, P, C, \delta)$ and Ω be the cyclic ordering of the vertices of $X \cap Y$ as they appear in C_1. Hence, $\delta = (\Gamma, \mathcal{D})$ is a rendition of $(G' = G[Y], \Omega)$. By Proposition 10.2.10, we can assume δ to be well-linked.

By Lemma 10.4.4 and Lemma 10.4.6, given that $r \geq \max\{a+3, k+1, 7\}$, there are two ground-maximal sphere decompositions $\delta' = (\Gamma', \mathcal{D}')$ and $\delta_v = (\Gamma_v, \mathcal{D}_v)$ of G' and $G - v$, respectively, that have property $\Pi_{\mathcal{H},k}$ and such that the $(r-2)$-central wall W' of W is grounded in both δ' and δ_v.

Then, by Lemma 10.2.15, there exists a ground-maximal sphere decomposition δ^* of G such that each cell δ^* is either a cell of δ' or a cell of δ_v, and is thus has property $\Pi_{\mathcal{H},k}$. Thus, by Lemma 10.4.3, G has $\mathcal{H}^{(a-1)}$-planar treewidth at most k. $\square$

10.4.2 The algorithm

In this section, we prove Lemma 10.1.7.

Let us first remark that we can use Courcelle's theorem (Proposition 4.3.2) to check whether a graph of bounded treewidth has $\mathcal{H}$-planar treewidth at most k.

Lemma 10.4.8. *If $\mathcal{H}$ is a CMSO-definable graph class, then having $\mathcal{H}$-planar treewidth at most k is expressible in CMSO logic.*

Proof. The class $\mathcal{PT}_k$ of graphs with planar treewidth at most k is minor-closed. Therefore, by Robertson and Seymour's seminal result, the number of obstructions of $\mathcal{PT}_k$ is finite.

As mentioned in Observation 10.0.1, $\mathsf{torso}(G, X)$ is expressible in CMSO logic. Moreover, as proven in [68, Subsection 1.3.1], the fact that a graph H is a minor of a graph G is expressible in CMSO logic, which implies that $\mathcal{PT}_k$ is expressible in CMSO logic. Indeed, it suffice to check for a graph G whether it contains or not the obstructions of $\mathcal{PT}_k$. $\square$

Proof of Lemma 10.1.7. We apply Theorem 10.2.2 to G, with $b = \lceil\sqrt{a+3}\rceil + 2$ and $r = \mathsf{odd}(\max\{a+3, k+1, 7\})$. It runs in time $\mathcal{O}_{k,a}(n+m)$.

If G has treewidth at most $f_{10.2.2}(b) \cdot r$, then we apply Proposition 4.3.2 to G in time $\mathcal{O}_{k,a}(n)$ and solve the problem. We can do so because the graphs in $\mathcal{H}^{(a-1)}$ have a bounded size, so $\mathcal{H}^{(a-1)}$ is a finite graph class, hence trivially CMSO-definable. Therefore, by Lemma 10.4.8, having $\mathcal{H}$-planar treewidth at most k is expressible in CMSO logic.

If G contains an apex grid of height b as a minor, then, given that $b \geq \sqrt{a+3}+2$, by Lemma 10.2.3, we obtain that G has no planar $\mathcal{G}^{(a-1)}$-modulator, where $\mathcal{G}$ is the class of all graphs. Hence, by Lemma 10.2.5, G has no sphere decomposition δ whose cells all have size at most $a-1$. This implies, by Lemma 10.4.4, that G has $\mathcal{H}^{(a-1)}$-planar treewidth at least $k+1$. Therefore, we report a no-instance.

Hence, we can assume that there is a flatness pair $(W, \mathfrak{R})$ of height r in G whose $\mathfrak{R}$-compass G' has treewidth at most $f_{10.2.2}(b) \cdot r$. Let v be a central vertex of W. We apply Proposition 4.3.2 to G' in time $\mathcal{O}_{k,a}(n)$ and we recursively apply our algorithm to $G - v$. If the outcome is a no-instance for one of them, then this is also a no-instance for G. Otherwise, the outcome is a yes-instance for both. Then, by Lemma 10.4.7, given that $r \geq \max\{a+3, k+1, 7\}$, we can return a yes-instance.

The running time of the algorithm is $T(n) = \mathcal{O}_{k,a}(n+m) + T(n-1) = \mathcal{O}_{k,a}(n(n+m))$. $\square$

10.5 Applications

This section contains several algorithmic applications of Theorem 2.6.1, as well as a discussion on similar applications for Theorem 2.6.2 and Theorem 2.6.3. To combine nice algorithmic properties of planar graphs and graphs from $\mathcal{H}$, we need a variant of Theorem 2.6.1 that guarantees the existence of a polynomial-time algorithm computing a planar $\mathcal{H}$-modulator in an $\mathcal{H}$-planar graph. (Let us remind that Theorem 2.6.1 only guarantees the existence of a polynomial-time algorithm deciding whether an input graph is $\mathcal{H}$-planar.) Thus, we start with the proof of the following theorem. The proof of the theorem combines self-reduction arguments with the hereditary properties of $\mathcal{H}$.

Theorem 10.5.1. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Then there exists a polynomial-time algorithm constructing for a given $\mathcal{H}$-planar graph G a planar $\mathcal{H}$-modulator.*

Proof. If $\mathcal{H}$ is trivial in the sense that $\mathcal{H}$ includes every graph, then we choose a planar $\mathcal{H}$-modulator $X = \{\emptyset\}$ of G. Otherwise, because $\mathcal{H}$ is hereditary, there is a graph F of the minimum size such that $F \notin \mathcal{H}$. We say that F is a *minimum forbidden subgraph*. Because the inclusion in $\mathcal{H}$ is decidable, F can be found in constant time by brute force checking all graphs of size $1, 2, \ldots$ where the constant depends on $\mathcal{H}$.

We consider two cases depending on whether F is connected or not.

Assume that F is connected. We pick an arbitrary vertex $r \in V(F)$ and declare it to be the *root* of F. We construct five copies $F_1, \ldots, F_5$ of F rooted in $r_1, \ldots, r_5$, respectively. Then we define F' to be the graph obtained from $F_1, \ldots, F_5$ by unifying their root into a single vertex r which is defined to be the root of F'. Our self-reduction arguments are based on the following claim.

Claim 10.5.2. *Let v be a vertex of a graph G. Then G has a planar $\mathcal{H}$-modulator X with $v \in X$ if and only if the graph G_v obtained from G and F' by unifying v and the root of F' has a planar $\mathcal{H}$-modulator.*

Proof of claim. Suppose that X is a planar $\mathcal{H}$-modulator of G with $v \in X$. Then each connected component of $G_v - X$ is either a connected component of $G - X$ or a connected component of one of the graphs $F_1 - r_1, \ldots, F_5 - r_5$. Because $F_i - r_i \in \mathcal{H}$ for $i \in [5]$, we have that each connected component of $G_v - X$ is in $\mathcal{H}$. Furthermore, the torsos of X with respect to G and G_v are the same because the root r of F' is the unique vertex of F' in X. Therefore, X is a planar $\mathcal{H}$-modulator of G_v.

Assume that G_v has a planar $\mathcal{H}$-modulator X. We show that $r = v \in X$. For the sake of contradiction, assume that $v \notin X$. Then because $F_1, \ldots, F_5 \notin \mathcal{H}$, for each $i \in [5]$, there is $x_i \in V(F_i)$ distinct from r_i such that $x_i \in X$. We pick each x_i to be a vertex in minimum distance from r_i in F_i. This means that each F_i contains a (x_i, r_i)-path P_i such that each vertex of $P_i - x_i$ is not in X. Then the vertices $\bigcup_{i=1}^{5}(V(P_i) \setminus \{x_i\})$ are in the same connected component of $G_v - X$. However, this means that $\{x_1, \ldots, x_5\}$ is a clique of size five in the torso of X contradicting planarity. Thus, $v \in X$. Because $v \in X$ and $v = r$ is the unique common vertex of G and F' in G_v, we have that $X' = X \setminus (V(F') \setminus \{r\})$ is a planar $\mathcal{H}$-modulator of G_v. This completes the proof of the claim. $\diamond$

By Claim 10.5.2, we can use self-reduction. First, we apply Theorem 2.6.1 to check in polynomial time whether G is a yes-instance of $\mathcal{H}$-PLANARITY. If not, then there is no planar $\mathcal{H}$-modulator. Otherwise, denote by $v_1, \ldots, v_n$ the vertices of G, set $G_0 = G$, and set $X := \emptyset$ initially. Then, for each $i := 1, \ldots, n$, we do the following:

- set G' to be the graph obtained from G_{i-1} and F' by unifying v_i and r,

- run the algorithm for $\mathcal{H}$-PLANARITY on G',
- if the algorithm returns a yes-answer then set $G_i = G'$ and $X := X \cup \{v_i\}$, and set $G_i = G_{i-1}$, otherwise.

Claim 10.5.2 immediately implies that X is a planar $\mathcal{H}$-modulator for G. Clearly, if $\mathcal{H}$-PLANARITY can be solved in polynomial time then the above procedure is polynomial. This concludes the proof for the case when F is connected.

Now, F is not connected. Our arguments are very similar to the connected case, and therefore, we only sketch the proof. Again, we construct five copies $F_1, \ldots, F_5$ of F. Then, we construct a new root vertex r and make it adjacent to every vertex of the copies of F. The following observation is in order.

Claim 10.5.3. *Let v be a vertex of a graph G. Then G has a planar $\mathcal{H}$-modulator X with $v \in X$ if and only if the graph G_v obtained from G and F' by unifying v and the root of F' has a planar $\mathcal{H}$-modulator.*

Proof of claim. Suppose that X is a planar $\mathcal{H}$-modulator of G with $v \in X$. Then each connected component of $G_v - X$ is either a connected component of $G - X$ or a connected component of one of the graphs $F_1, \ldots, F_5$. Because each connected component of F_i is in $\mathcal{H}$ for $i \in [5]$ by the minimality of F, we have that each connected component of $G_v - X$ is in $\mathcal{H}$. Also, the torsos of X with respect to G and G_v are the same because r is the unique vertex of F' in X. Thus, X is a planar $\mathcal{H}$-modulator of G_v.

Assume that G_v has a planar $\mathcal{H}$-modulator X. We show that $r = v \in X$. For the sake of contradiction, assume that $v \notin X$. Then because $F_1, \ldots, F_5 \notin \mathcal{H}$, for each $i \in [5]$, there is $x_i \in F_i$ distinct from r_i such that $x_i \in X$. Since each x_i is adjacent to $r \notin X$, $\{x_1, \ldots, x_5\}$ is a clique of size five in the torso of X contradicting planarity. Thus, $v \in X$. Because $v \in X$ and $v = r$ is the unique common vertex of G and F' in G_v, $X' = X \setminus \bigcup_{i=1}^{5} V(F_i)$ is a planar $\mathcal{H}$-modulator of G_v. This completes the proof of the claim. $\diamond$

Claim 10.5.3 immediately implies that we can apply the same self-reduction procedure as for the connected case. This concludes the proof. $\square$

We remark that similar arguments can be used to construct decompositions for graphs with $\mathcal{H}$-planar treewidth of treedepth at most k. We sketch the algorithms in the following corollaries.

Corollary 10.5.4. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, polynomial-time decidable, and closed under disjoint union graph class. Suppose that there is an* FPT *algorithm solving* VERTEX DELETION TO $\mathcal{H}$ *parameterized by the solution size. Then there exists a polynomial-time algorithm constructing, for a given graph G with $\mathcal{H}$-planar treewidth at most k, a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator S of a graph G and a tree decomposition $\mathcal{T}$ of $\mathsf{torso}(G,S)$ of planar width at most k.*

Sketch of the proof. We follow the proof of Theorem 10.5.1. The problem is trivial if $\mathcal{H}$ includes every graph. Otherwise, because $\mathcal{H}$ is hereditary, there is a minimum forbidden subgraph F. We use F to identify a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator. We have two cases depending on whether F is connected or not. In this sketch, we consider only the connected case; the second case is analyzed in the same way as in the proof of Theorem 10.5.1. We pick an arbitrary vertex $r \in V(F)$ and declare it to be the *root* of F. We construct $h = \max\{5, k+2\}$ copies $F_1, \ldots, F_h$ of F rooted in $r_1, \ldots, r_h$, respectively. Then we define F' to be the graph obtained from $F_1, \ldots, F_h$ by unifying their root into a single vertex r which is defined to be the root of F'. Then similarly to Claim 10.5.2, we have the following property for every $v \in V(G)$: G has a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator S with $v \in S$ if and only if the graph G_v

obtained from G and F' by unifying v and the root of F' has a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator S. Thus, we can find S by using self-reduction by calling the algorithm from Theorem 2.6.3.

In the next step, we again use self-reducibility to construct a tree decomposition $\mathcal{T}$ of $G' = \mathsf{torso}(G, S)$ of planar width at most k. For this, we consider all pairs $\{u, v\}$ of non-adjacent vertices of G'. For each $\{u, v\}$, we add the edge uv to the considered graph, and then use the algorithm from Theorem 2.6.3 to check whether the obtained graphs admits a tree decomposition of planar width at most k (in this case, $\mathcal{H}$ contains the unique empty graph). If yes, we keep the edge uv, and we discard the pair $\{u, v\}$, otherwise. Let G^* be the graph obtained as the result of this procedure. We have that G^* has a tree decomposition of planar width at most k where every bag of size at most $k + 1$ is a clique. Furthermore, the adhesion sets for every bag of size at least $k + 2$, whose torso is planar, are cliques of size at most 4. Then we can find all such bags by decomposing G^* via clique-separators of size at most 4. After that, we can find the remaining bags using the fact that they are cliques. This completes the sketch of the proof. □

Corollary 10.5.5. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, polynomial-time decidable, and closed under disjoint union graph class. Suppose that there is an* FPT *algorithm solving* Vertex Deletion to $\mathcal{H}$ *parameterized by the solution size. Then there exists a polynomial-time algorithm constructing, for a given graph G with $\mathcal{H}$-planar treedepth at most k, a certifying elimination sequence $X_1, \ldots, X_k$.*

Sketch of the proof. We again follow the proof of Theorem 10.5.1 and use self-reduction. Assume that $\mathcal{H}$ does not include every graph and let F be a rooted minimum forbidden subgraph. Assume that F is connected; the disconnected case is analyzed similarly to the proof of Theorem 10.5.1. We construct the graph F' as follows:

- construct a copy of the complete graph K_{4k}.
- for every vertex v of the complete graph, construct five copies $F_1, \ldots F_5$ of F rooted in $r_1, \ldots, r_5$, respectively, and then identify $r_1, \ldots, r_5$ with v.

Then we declare an arbitrary vertex r of the complete graph in F' to be its root. Then similarly to to Claim 10.5.2, we have the following property for every $v \in V(G)$: G admits a certifying elimination sequence $X_1, \ldots, X_k$ with $v \in X_1$ if and only if the graph G' obtained from G and F' by unifying v and the root of F' has $\mathcal{H}$-planar treedepth at most k. We use this observation to identify X_1. Then we find $X_2, \ldots, X_k$ by using the same arguments inductively. This concludes the sketch of the proof. □

Now, we can proceed to algorithmic applications of Theorem 10.5.1, Corollary 10.5.4, and Corollary 10.5.5.

10.5.1 Colourings

Our first algorithmic application is the existence of a polynomial-time additive-approximation algorithm for graph coloring on $\mathcal{H}$-planar graphs.

Theorem 10.5.6. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that there is a polynomial-time algorithm computing the chromatic number $\chi(H)$ for $H \in \mathcal{H}$. Then, there exists a polynomial-time algorithm that, given an $\mathcal{H}$-planar graph G, produces a proper coloring of G using at most $\chi(G) + 4$ colors.*

For example, the class of perfect graphs is hereditary, CMSO-definable, and polynomial-time decidable [62]. Moreover, the chromatic number of a perfect graph is computable in polynomial time [155]. Thus by Theorem 10.5.6, when $\mathcal{H}$ is the class of perfect graphs, for any $\mathcal{H}$-planar graph G, there is a polynomial-time algorithm coloring G in at most $\chi(G)+4$ colors.

Proof. By Theorem 10.5.1, there is a polynomial-time algorithm computing a planar $\mathcal{H}$-modulator S of an $\mathcal{H}$-planar graph G. Given that $G[S]$ is planar, there is a proper coloring of $G[S]$ with colors $[1,4]$ by the Four Color theorem [15, 257]. Moreover, the proof of the Four Color theorem is constructive and yields a polynomial-time algorithm producing a 4-coloring of a planar graph. For each component $C \in \mathsf{cc}(G-S)$, by the assumptions of the theorem, we can compute $\chi(G[C]) \leq \chi(G)$ in polynomial time. Since all components are disjoint, we use at most $\chi(G)$ to color all the components of $\mathsf{cc}(G-S)$ and then additional four colors to color $G[S]$. This gives a proper $(\chi(G)+4)$-coloring of G. □

For a graph with bounded $\mathcal{H}$-planar treedepth, we can use Corollary 10.5.5 to find a certifying elimination sequence. Then by repetitive applications of Theorem 10.5.6, we immediately obtain a bound on the chromatic number.

Corollary 10.5.7. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that there is a polynomial-time algorithm computing the chromatic number $\chi(H)$ for $H \in \mathcal{H}$. Then, there exists a polynomial-time algorithm that, given a graph G with $\mathcal{H}\text{-}\mathsf{ptd}(G) \leq k$, produces a proper coloring of G using at most $\chi(G)+4k$ colors.*

We can also prove a similar result for graphs of bounded $\mathcal{H}$-planar treewidth.

Theorem 10.5.8. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that there is a polynomial-time algorithm computing the chromatic number $\chi(H)$ for $H \in \mathcal{H}$. Then, there exists a polynomial-time algorithm that, given a graph G with $\mathcal{H}\text{-}\mathsf{ptw}(G) \leq k$, produces a proper coloring of G using at most $\chi(G)+\max\{4,k+1\}$ colors.*

Proof. Given a graph G, we use the algorithm from Corollary 10.5.4 to find a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator S of G with a tree decomposition $\mathcal{T}$ of $\mathsf{torso}(G,S)$ of planar width at most k. As in the proof of Theorem 10.5.6, for each component $C \in \mathsf{cc}(G-S)$, by the assumptions of the theorem, we can compute $\chi(G[C]) \leq \chi(G)$ in polynomial time. Let us now find a proper coloring of $\mathsf{torso}(G,S)$ with $\max\{4,k+1\}$ colors. For this, it is enough to find a proper coloring of the torso of each bag of $\mathcal{T}$ with at most $\max\{4,k+1\}$ colors. If a bag as at most $k+1$ vertices, then $k+1$ colors suffices, and if a bag has a planar torso, then four colors are enough [15, 257], hence the result. □

10.5.2 Counting perfect matchings

Our second example of applications of Theorem 2.6.1 concerns counting perfect matchings. While counting perfect matchings on general graphs is #P-complete, on planar graphs, it is polynomial-time solvable by the celebrated Fisher–Kasteleyn–Temperley (FKT) algorithm [118, 182, 300]. Given a graph G with an edge weight function $w : E(G) \to \mathbb{N}$, the weighted number of perfect matchings is

$$\mathsf{pmm}(G) = \sum_{M} \prod_{uv \in M} w(uv)$$

where the sum is taken over all perfect matchings M.

Theorem 10.5.9. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that the weighted (resp. unweighted) number of perfect matchings $\mathsf{pmm}(H)$ is computable in polynomial time for each graph H in $\mathcal{H}$. Then, there exists a polynomial-time algorithm that, given a weighted (resp. unweighted) $\mathcal{H}$-planar graph G, computes its weighted number of perfect matchings $\mathsf{pmm}(G)$.*

Examples of classes of graphs $\mathcal{H}$ where counting perfect matching can be done in polynomial time are graphs excluding a shallow-vortex as a minor [303] (see also [18, 70, 72, 105, 135]), bounded clique-width graphs [71], and chain, co-chain, and threshold graphs [245].

To prove this, we use Valiant "matchgates" [308], or more precisely the gadgets from [298].

Proposition 10.5.10 ([298])**.** *Let G be a graph and (A, B) be a separation of G of order two (resp. three) with $A \cap B = \{a, b\}$ (resp. $A \cap B = \{a, b, c\}$). Denote by G_B the graph obtained from $G[B]$ by removing the edges with both endpoints in $A \cap B$. Let also $p_\gamma := \mathsf{pmm}(G_B - \gamma)$ for each $\gamma \subseteq A \cap B$. Then, $\mathsf{pmm}(G) = \mathsf{pmm}(G')$, where G' is the graph obtained from $G[A]$ by adding the corresponding planar gadget of Figure 10.10 (that depends on the size of $A \cap B$ and the parity of B).*

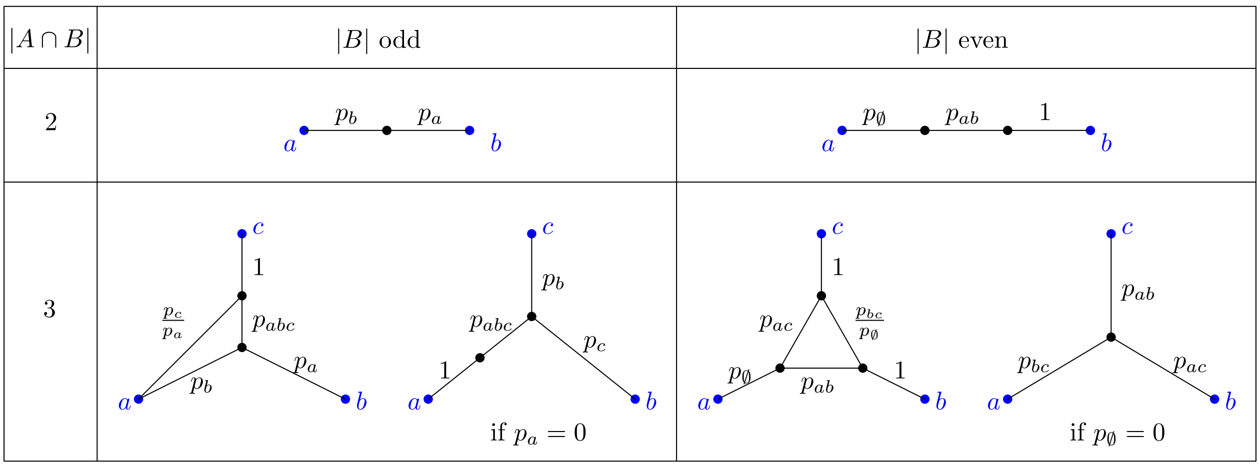

Figure 10.10: The gadget (from [298]) we replace G_B with, depending on the size of $A \cap B$ and the parity of $|B|$; to simplify notation, we write the indices of the weight p_γ as the lists of the elements of γ (e.g., we write p_{abc} instead of $p_{\{a,b,c\}}$). Note that, if $p_a = 0$ or $p_\emptyset = 0$, then there may be a variant gadget.

We now prove Theorem 10.5.9.

Proof of Theorem 10.5.9. Let G be an $\mathcal{H}$-planar graph and $w : E(G) \to \mathbb{N}$ be a weight function (with $w = 1$ in the unweighted case).

If G is not connected, then the weighted number of perfect matchings in G is the product of the number of perfect matchings in the connected components of G. Hence, we assume without loss of generality that G is connected.

Obviously, if G has an odd number of vertices, then $\mathsf{pmm}(G) = 0$. So we assume that $|V(G)|$ is even. Additionally, if G has a cut vertex v, let $C_1 \in \mathsf{cc}(G - v)$ and $C_2 = \bigcup_{C \in \mathsf{cc}(G-v) \setminus \{C_1\}} C$. Given that $|V(G - v)|$ is odd, exactly one of C_1 and C_2, say C_1, has an odd number of vertices. Then the (weighted) number of perfect matchings is the product of $\mathsf{pmm}(C_1 \cup v)$ and $\mathsf{pmm}(C_2)$. Therefore, we can assume without loss of generality that G is 2-connected.

By Theorem 10.5.1, there is a polynomial-time algorithm computing a planar $\mathcal{H}$-modulator S of an $\mathcal{H}$-planar graph G. Let δ' be a sphere embedding of $\mathsf{torso}(G, S)$. Let δ be the sphere embedding of $G[S]$ obtained from δ' by removing the edges that do not belong to $G[S]$. By Claim 10.2.7, for every $D \in \mathsf{cc}(G - S)$, there is a δ-aligned disk Δ_D such that:

- the vertices of $N_G(V(D))$ are in the disk Δ_D, i.e. $N_G(V(D)) \subseteq \pi_\delta(N(\delta) \cap \Delta_D)$,

- with all but at most one (in the case $|N_G(V(D))| \leq 4$) being exactly the vertices of the boundary of Δ_D, i.e. there is a set $X_D \subseteq N_G(V(D))$ of size $\min\{|N_G(V(D))|, 3\}$ such that $X_D = \pi_\delta(\mathsf{bd}(\Delta_D) \cap N(\delta))$.

Actually, Claim 10.2.7 is proved for δ', but it is still true after removing edges to obtain δ. Note that, given that G is 2-connected, we have $2 \leq |X_D| \leq 3$.

Let $\{\Delta_1, \dots, \Delta_p\}$ be the set of all such disks Δ_D, ordered so that there is no $i < j \in [p]$ with $\Delta_i \supsetneq \Delta_j$. Note that for any D, D' such that $X_D = X_{D'}$, we assume that $\Delta_D = \Delta'_D$, and thus, that there is a unique i such that $\Delta_i = \Delta_D = \Delta'_D$. Given that there are at most $\binom{n}{3}$ separators in G, we have $p \leq n^3$.

For $i \in [p]$, let $Z_i := \{C \in \mathsf{cc}(G - S) \mid N_G(V(C)) \subseteq \pi_\delta(N(\delta) \cap \Delta_i)\}$ be the set of connected components whose neighborhood is in Δ_i. For $i \in [p]$ in an increasing order, we will construct by induction a tuple $(G_i, S_i, \delta_i, w_i)$ such that

1. S_i is a planar $\mathcal{H}$-modulator of the graph G_i with $\mathsf{cc}(G_i - S_i) = \mathsf{cc}(G - S) \setminus \bigcup_{j \leq i} Z_j$,
2. δ_i is a sphere embedding of $G_i[S_i]$ with $\delta_i \setminus \bigcup_{j \leq i} \mathsf{int}(\Delta_j) = \delta \setminus \bigcup_{j \leq i} \mathsf{int}(\Delta_j)$, and
3. $w_i : E(G_i) \to \mathbb{N}$ is a weight function such that $\mathsf{pmm}(G_i) = \mathsf{pmm}(G)$.

Therefore, $\mathsf{cc}(G_p - S_p) = \emptyset$, so $G_p = G[S_p]$ is a planar graph with $\mathsf{pmm}(G_p) = \mathsf{pmm}(G)$. Then, by [183], $\mathsf{pmm}(G_p)$, and thus $\mathsf{pmm}(G)$, can be computed in polynomial time.

Obviously, the conditions are originally respected for $(G_0, S_0, \delta_0, w_0) = (G, S, \delta, w)$. Let $i \in [p]$. Suppose that we constructed $(G_{i-1}, S_{i-1}, \delta_{i-1}, w_{i-1})$. Let $B_i := V(\mathsf{inner}_\delta(\Delta_i)) \cup V(Z_i)$ be the set of vertices that are either in Δ_i or in a connected component whose neighborhood is in Δ_i. Let also $X_i := \pi_\delta(N(\delta_{i-1}) \cap \Delta_i)$ be the vertices on the boundary of Δ_i. Given that there are no $j < i$ such that $\Delta_i \subseteq \Delta_j$, and by the induction hypothesis (2), there is $D \in \mathsf{cc}(G - S)$ such that $X_i = X_D$, so we have $2 \leq |X_i| \leq 3$. If $|X_i| = 2$, we label its elements a and b, and if $|X_i| = 3$, we label its elements a, b, c. Let H_i be the graph induced by B_i but where we remove the edges whose endpoints are both in X_i.

For $\gamma \subseteq X_i$, let

$$p_\gamma := \mathsf{pmm}(H_i - \gamma).$$

Then, by Proposition 10.5.10, if we manage to compute p_γ for each $\gamma \subseteq X_i$, then we can replace H_i with the corresponding planar gadget F_i of Figure 10.10 (that depends on the size of X_i and the parity of $|V(H_i)| = |B_i|$) to obtain G_i and the weight function w_i, so that $\mathsf{pmm}(G_i) = \mathsf{pmm}(G_{i-1}) = \mathsf{pmm}(G)$. So Item (3) of the induction holds. Then $S_i := S_{i-1} \setminus Z_i \cup V(F_i)$ is a planar $\mathcal{H}$-modulator of G_i with $\mathsf{cc}(G_i - S_i) = \mathsf{cc}(G_{i-1} - S_{i-1}) \setminus Z_i = \mathsf{cc}(G - S) \setminus \bigcup_{j \leq i} Z_j$. So Item (1) of the induction holds. Additionally, δ_i, that is obtained from δ_{i-1} by adding the gadget to the planar embedding, is such that $\delta_i \setminus \mathsf{int}(\Delta_i) = \delta_{i-1} \setminus \mathsf{int}(\Delta_i)$, and thus respects Item (2) by induction hypothesis. Then, the obtained tuple $(G_i, S_i, \delta_i, w_i)$ would respects the induction hypothesis.

What is left is to explain how to compute p_γ for $\gamma \subseteq X_i$. For each $v \in X_i \setminus \gamma$, we guess with which vertex of $B_i \setminus X_i$ v is matched. There are $\mathcal{O}(n)$ choices for each such v, and therefore $\mathcal{O}(n^3)$ guesses to match all vertices in $X_i \setminus \gamma$. Let H'_i be the graph obtained from H_i after removing X_i and the vertices matched with vertices of $X_i \setminus \gamma$. We have $p_\gamma = \mathsf{pmm}(H'_i) + |X_i \setminus \gamma|$. Note that for any $C \in Z_i$, $X_i \subseteq N_{G_{i-1}}(C) = N_G(V(C))$. Indeed, otherwise, there would be $j > i$ such that $\Delta_C = \Delta_j$, with $\Delta_j \subsetneq \Delta_i$, a contradiction. Therefore, for any $C \in Z_i$, if $|N_G(V(C))| \leq 3$, then C is a connected component of H'_i, and if $|N_G(V(C))| = 4$, then C is a block of H'_i. The rest of H'_i is $\mathsf{inner}_{\delta_{i-1}}(\Delta_i)$, that is a planar graph. Therefore, H'_i is a graph whose blocks are either planar or in $\mathcal{H}$. As said above, for each cut vertex, we know in which block it should be matched depending on the parity of the blocks. Hence, we can compute the weighted number of perfect matchings in each block separately. Note that the weights added by the gadgets of Figure 10.10 are only present in planar blocks, where we know how to compute the weighted number of perfect matchings in polynomial time. In blocks belonging to $\mathcal{H}$, the weight of edges did not change, so in the unweighted (resp. weighted) case, we know how to compute the unweighted (resp. weighted) number of perfect matchings in polynomial time. Thus, we can compute $\mathsf{pmm}(H_i')$ and, therefore, p_γ in polynomial time. □

Again, by repetitive applications of Theorem 10.5.9, we immediately obtain the following for graphs with bounded $\mathcal{H}$-planar treedepth.

Corollary 10.5.11. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that the weighted (resp. unweighted) number of perfect matchings $\mathsf{pmm}(H)$ is computable in polynomial time for graphs in $\mathcal{H}$. Then, there exists an algorithm that, given a weighted (resp. unweighted) graph G with $\mathcal{H}$-$\mathsf{ptd}(G) \leq k$, computes its weighted (resp. unweighted) number of perfect matchings $\mathsf{pmm}(G)$ in time $n^{O(k)}$.*

If G has $\mathcal{H}$-planar treewidth at most k, even given a $\mathcal{P}^k \triangleright \mathcal{H}$-modulator of G, computing the number of perfect matchings is more difficult. Still, combining Theorem 10.5.9 with the dynamic programming approach of [303], it is easy to derive an algorithm that, given a weighted graph, computes the weighted number of its perfect matchings in time $n^{O(k)}$.

Theorem 10.5.12. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that the weighted (resp. unweighted) number of perfect matchings $\mathsf{pmm}(H)$ is computable in polynomial time for graphs in $\mathcal{H}$. Then, there exists an algorithm that, given $\mathcal{H}$-$\mathsf{ptw}(G) \leq k$, computes its weighted (resp. unweighted) number of perfect matchings $\mathsf{pmm}(G)$ in time $n^{O(k)}$.*

10.5.3 EPTAS for Independent Set

Baker's technique [20] provides PTAS and EPTAS for many optimization problems on planar graphs. This method is based on the fact that planar graphs have bounded *local treewidth*. In particular, for every planar graph G and every vertex $v \in V(G)$, $\mathsf{tw}(N_G^r[v]) = \mathcal{O}(r)$, where $N_G^r[v]$ is the *closed r-neighborhood* of v, that is, the set of vertices at distance at most r from v (see [259]).

Given that the torso of a planar $\mathcal{H}$-modulator X is planar, we extend Baker's technique to $\mathcal{H}$-planar graphs, by replacing the role of treewidth with *$\mathcal{H}$-treewidth*. As mentioned in the introduction, $\mathcal{H}$-treewidth has been defined by Eiben, Ganian, Hamm, and Kwon [104]. Its algorithmic properties have been studied in [6, 104, 174].

Let us explain how the EPTAS works on Independent Set. The problem of Independent Set asks for an *independent set* of maximum size, that is a set S such that no edge of G has both endpoints in S. Let us remind that an independent set in a graph G is a set of pairwise nonadjacent vertices. We use $\alpha(G)$ to denote the maximum size of an independent set of G. The following is proved in [174, Theorem 3.16] (it is proven for the dual problem of Vertex Cover).

Proposition 10.5.13 ([174]). *Let $\mathcal{H}$ be a hereditary graph class on which* Independent Set *is polynomial-time solvable. Then* Independent Set *can be solved in time $2^k \cdot n^{\mathcal{O}(1)}$ when given a tree $\mathcal{H}$-decomposition of width at most $k-1$ consisting of $n^{\mathcal{O}(1)}$ nodes.*

With Theorem 10.5.1 on hands, the proof of the following theorem is almost identical to Baker's style algorithms for planar graphs. The main observation is that the supergraph of an input $\mathcal{H}$-planar graph G obtained by adding to G all edges of the torso of its planar modulator, is of bounded local $\mathcal{H}$-treewidth.

Theorem 10.5.14. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. We also assume that there is a polynomial-time algorithm computing a maximum independent set of graphs in $\mathcal{H}$. Then, there is an algorithm that, given $\varepsilon > 0$ and an $\mathcal{H}$-planar graph G, computes in time $2^{\mathcal{O}(1/\varepsilon)} \cdot |V(G)|^{\mathcal{O}(1)}$ an independent set of G of size at least $(1-\varepsilon)\cdot\alpha(G)$.*

Examples of graph classes where $\alpha(G)$ is computable in polynomial time are perfect graphs [155] or graphs excluding P_6 as an induced subgraph [157]. Theorem 10.5.14 could also be modified for graph classes where computing a maximum independent set is quasi-polynomial or subexponential. In these cases, the approximation ratio will remain $1-\varepsilon$, but the approximation algorithm's running time will also become quasi-polynomial or subexponential.

Proof. Let G be an $\mathcal{H}$-planar graph. We assume without loss of generality that G is a connected graph and $\varepsilon < 1$. By Theorem 10.5.1, there is a polynomial-time algorithm computing a planar $\mathcal{H}$-modulator X of G. Let v be an arbitrary vertex in X. For $i \in \mathbb{N}$, let L_i denote the set of vertices of X at distance i of v in $\mathsf{torso}(G,X)$. Observe that the endpoints of an edge in $\mathsf{torso}(G,X)$ belong to at most two (consecutive) L_is. Given that, for $C \in \mathsf{cc}(G-X)$, $N_G(V(C))$ induces a clique in $\mathsf{torso}(G,X)$, it therefore implies that the vertices of $N_G(V(C))$ belong to at most two (consecutive) L_is.

Let k be the smallest integer such that $2/k \le \varepsilon$. For $i \in [0,k-1]$, let S_i be the union of the L_j for which j is equal to $i \pmod k$. Let $X_i := X \setminus S_i$ and let $\mathcal{C}_i$ be the set of components $C \in \mathsf{cc}(G-X)$ such that $N_G(V(C)) \subseteq X_i$. Let G_i be the induced subgraph of G induced by X_i and $\mathcal{C}_i$. Given that the connected components of $G_i - X_i$ are connected components of $G-X$, it implies that $\mathsf{torso}(G_i,X_i)$ is a subgraph of $\mathsf{torso}(G,X) - S_i$, and is thus planar. Therefore, as proven in [74, Corollary 7.34] (see also [259, (2.5)]), $\mathsf{torso}(G_i,X_i)$ has treewidth $\mathcal{O}(k)$ and there is a polynomial-time algorithm constructing a tree decomposition (T_i,β_i) of $\mathsf{torso}(G_i,X_i)$ of width $\mathcal{O}(k)$. Therefore, $(T_i,\beta_i,\mathcal{C}_i)$ is a tree $\mathcal{H}$-decomposition of $\mathsf{torso}(G_i,X_i) \cup G[\mathcal{C}_i]$, and hence of G_i, of width $\mathcal{O}(k)$. We apply Proposition 10.5.13 to find in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)} = 2^{\mathcal{O}(1/\varepsilon)} \cdot n^{\mathcal{O}(1)}$ a maximum independent set I_i of G_i. Note that I_i is also an independent set of G. We return as a solution the maximum size set I_i for $i \in [0,k-1]$.

Let I^* be a maximum independent set of G, of size $|I^*| = \alpha(G) = \mathrm{OPT}$. For $i \in [0,k-1]$, let $\mathcal{D}_i := \mathsf{cc}(G-X) \setminus \mathcal{C}_i$ and let $S_i' := S_i \cup V(\mathcal{D}_i)$. Let $C \in \mathsf{cc}(G-X)$. Given that the vertices of $N_G(V(C))$ belong to at most two (consecutive) L_is, it implies that C belongs to at most two (consecutive) $\mathcal{D}_i$s. Additionally, the S_is partition X. Therefore,

$$\sum_{i=0}^{k-1} |S_i'| = |X| + \sum_{i=0}^{k-1} |V(\mathcal{D}_i)| \le |X| + 2|V(G)\setminus X| = 2|V(G)| - |X| \le 2|V(G)|.$$

Hence, there is $j \in [0,k-1]$ such that $|I^* \cap S_j'| \le 2|I^*|/k$. Since $I^* \setminus S_j'$ is an independent set in G_j, we have the following:

$$|I_j| \ge |I^* \setminus S_j'| = |I^*| - |I^* \cap S_j'| \ge (1-2/k)|I^*| \ge (1-\varepsilon)\mathrm{OPT}.$$

Thus, the algorithm returns a solution of size at least $(1-\varepsilon)\mathrm{OPT}$ in time $2^{\mathcal{O}(1/\varepsilon)} \cdot n^{\mathcal{O}(1)}$. □

As with the previous applications, it is possible to generalize the technique for the graphs of bounded $\mathcal{H}$-planar treewidth or treedepth. We sketch our algorithms in the following two corollaries.

Corollary 10.5.15. *Let $\mathcal{H}$ be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that the maximum independent set is computable in polynomial time for graphs in $\mathcal{H}$. Then, there exists a polynomial-time algorithm that, given $\varepsilon > 0$ and a graph G with*

$\mathcal{H}$-$\mathsf{ptd}(G) \le k$, *computes in time* $2^{\mathcal{O}(\max\{k,1/\varepsilon\})}|V(G)|^{\mathcal{O}(1)}$ *an independent set of* G *of size at least* $(1-\varepsilon)\cdot\alpha(G)$.

Sketch of the proof. We assume without loss of generality that G is a connected graph and $\varepsilon < 1$. We use Corollary 10.5.4 to find a $\mathcal{PT}_k \triangleright \mathcal{H}$-modulator S of G with a tree decomposition $\mathcal{T}$ of $\mathsf{torso}(G,S)$ of planar width at most k. Let $X_1,\ldots,X_\ell$ be the bags of the decomposition such that $G_i = \mathsf{torso}(G,X_i)$ is a planar graph for $i \in [\ell]$. Because G is connected, each G_i is a connected graph. We select arbitrarily ℓ vertices $v_i \in X_i$ for $i \in [\ell]$. For each $i \in [\ell]$ and every integer $j \ge 0$, we denote by L^i_j the set of vertices of X_i at distance j from v_i in G_i. Let $h \ge 2$ be the smallest integer such that $2/h \le \varepsilon$. For $j \in [0,h-1]$, let S_j be $\bigcup_p \bigcup_{i=1}^{\ell} L^i_p$ where the first union is taken over all $p \ge 0$ for which p is equal to $i \pmod h$. For each $j \in [0,h-1]$, let $Y_j = X \setminus S_j$. The crucial observation is that, because each G_i is planar, the adhesion set of X_i with other bags in $\mathcal{T}$ is in at most two consecutive sets L^i_q and L^i_{q+1} for some $j \ge 0$. This implies that $\mathsf{tw}(G[Y_i]) = \mathcal{O}(\max\{h,k\})$. This implies that we can apply the same Baker's style arguments as in the proof of Theorem 10.5.14. This concludes the sketch of the proof. ☐

Corollary 10.5.16. *Let* $\mathcal{H}$ *be a hereditary, CMSO-definable, and polynomial-time decidable graph class. Moreover, assume that the maximum independent set is computable in polynomial time for graphs in* $\mathcal{H}$. *Then, there exists a polynomial-time algorithm that, given* $\varepsilon > 0$, *a graph* G *with* $\mathcal{H}$-$\mathsf{ptw}(G) \le k$, *computes in time* $2^{\mathcal{O}(k+1/\varepsilon)}|V(G)|^{\mathcal{O}(1)}$ *an independent set of* G *of size at least* $(1-\varepsilon)\cdot\alpha(G)$.

Sketch of the proof. The main idea is the same as in Corollary 10.5.15. We assume without loss of generality that G is a connected graph and $\varepsilon < 1$. We use Corollary 10.5.5 to find a certifying elimination sequence $X_1,\ldots,X_k$. For $i \in [k]$ and $j \in [\ell_i]$, denote by $Y_{ij} \subseteq X_i$ the inclusion maximal subsets of X_i such that $G_{ij} = \mathsf{torso}(G - \cup_{h=1}^{i-1} X_h, Y_{ij})$ is connected. We arbitrarily select vertices $v_{ij} \in Y_{ij}$ for $i \in [k]$ and $j \in [\ell_i]$. For $i \in [k]$, $j \in [\ell_i]$, and each integer $h \ge 0$, denote by L^{ij}_h the set of vertices of Y_{ij} at distance h from v_{ij} in G_{ij}. Let $s \ge 2$ be the smallest integer such that $2/s \le \varepsilon$. For $t \in [0,s-1]$, let S_t be $\bigcup_p \bigcup_{i=1}^{k} \bigcup_{j=1}^{\ell_i} L^{ij}_p$ where the first union is taken over all $p \ge 0$ for which p is equal to $i \pmod s$. For each $t \in [0,s-1]$, let $Z_t = X \setminus S_t$. Now we have that $\mathsf{tw}(G[Z_t]) = \mathcal{O}(s+k)$. Using the same arguments as in Theorem 10.5.14 and Corollary 10.5.15, we show the desired approximation. This concludes the sketch of the proof. ☐

Baker's style arguments and other approaches apply to many optimization problems on planar graphs. For instance, dynamic programming algorithms on graphs of bounded $\mathcal{H}$-treewidth, as the one of Proposition 10.5.13, exist for other problems (see [104, 174]) that are amenable to Baker's technique. Investigating which of these problems admit good approximations on $\mathcal{H}$-planar graphs and whether there are meta-algorithmic theorems in the style of [82, 127, 149], is an interesting research direction that goes beyond the scope of this thesis.

10.6 Necessity of conditions

In our algorithmic results, we require that the considered graph classes $\mathcal{H}$ should satisfy certain properties. In this section, we discuss tightness of these properties. First, we show that the heredity condition for $\mathcal{H}$ is crucial for the existence of a polynomial-time algorithm for $\mathcal{H}$-Planarity. For this, we prove that $\mathcal{H}$-Planarity is NP-hard even if $\mathcal{H}$ consists of a single graph.

Theorem 10.6.1. *Let* $\mathcal{H}$ *be the class consisting of the complete graph on four vertices. Then* $\mathcal{H}$-Planarity *is* NP-*complete.*

Proof. We show NP-hardness by reducing from a variant of the PLANAR SAT problem. Consider a Boolean formula φ in the conjunctive normal form on n variables $x_1, \ldots, x_n$ with clauses $C_1, \ldots, C_m$. We define the graph $G(\varphi)$ with $2n + m$ vertices constructed as follows:

- for each $i \in [n]$, construct two vertices x_i and $\overline{x}_i$ and make them adjacent,
- for each $j \in [m]$, construct a vertex C_j,
- for each $i \in [n]$ and $j \in [m]$, make x_i and C_j adjacent if the clause C_j contains the literal x_i, and make $\overline{x}_i$ and C_j adjacent if C_j contains the negation of x_i.

It is known that the SAT problem is NP-complete when restricted to instances where (i) $G(\varphi)$ is planar, (ii) each variable occurs at most 4 times in the clauses—at most two times in positive and at most two times with negations, and (iii) each clause contains either two or three literals [206].

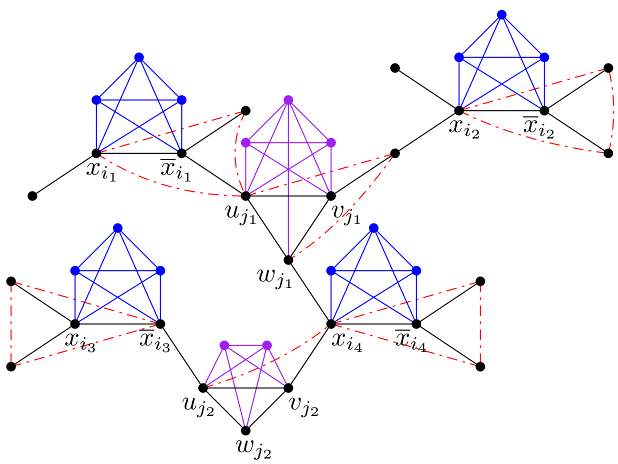

Figure 10.11: Construction of G' and G''. The vertices and edges of G' are shown in black and the additional vertices and edges of G'' are blue (for the variable) and purple (for the clauses). The construction is shown for four variables $x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}$ and two clauses $\overline{x}_1 \vee x_{i_2} \vee x_{i_4}$ and $\overline{x}_{i_3} \vee x_{i_4}$. Here, x_{i_1}, x_{i_3} occur in three clauses and x_{i_2}, x_{i_4} occur in four clauses. We also show in dashed red the edges of the torso of X constructed for the assignment $x_{i_1} = x_{i_2} = x_{i_4} = \mathit{true}$ and $x_{i_3} = \mathit{false}$.

Consider such an instance φ of PLANAR SAT with the variables $x_1, \ldots, x_n$ and the clauses $C_1, \ldots, C_m$. We construct the following graph G':

- for each $i \in [n]$, construct two *variable* vertices x_i and $\overline{x}_i$ and make them adjacent,
- for each $j \in [m]$, construct three *clause* vertices u_j, v_j, w_j and make them pairwise adjacent,
- for each $i \in [n]$ and $j \in [m]$, make x_i adjacent to one of the vertices u_j, v_j, w_j if the clause C_j contains the literal x_i, and make $\overline{x}_i$ adjacent to one of the vertices u_j, v_j, w_j if C_j contains the negation of x_i; we select the neighbor of x_i or $\overline{x}_i$ among u_j, v_j, w_j in such a way that no clause vertex is adjacent to two variable vertices.

Notice that because $G(\varphi)$ is planar, G' is a planar graph. In the next stage of our reduction, we construct G'' from G':

- for each $i \in [n]$, construct three vertices y_i^1, y_i^2, y_i^3 and make them pairwise adjacent and adjacent to both x_i and $\overline{x}_i$,

- for each $j \in [m]$ such that $|C_j| = 3$, construct a set of three vertices $Z_j = \{z_j^1, z_j^2, z_j^3\}$ and make the vertices of Z_j pairwise adjacent and adjacent to u_j, v_j, w_j,
- for each $j \in [m]$ such that $|C_j| = 2$, construct a set of two vertices $Z_j = \{z_j^1, z_j^2\}$ and make the vertices of Z_j adjacent and adjacent to u_j, v_j, w_j.

The construction of G' and G'' is shown in Figure 10.11. In our construction, we assume that each clause vertex w_j constructed for C_j of size two is not adjacent to any variable vertex.

We claim that φ has a satisfying assignment if and only if G'' has a planar $\mathcal{H}$-modulator.

Suppose that φ has a satisfying assignment and the variables $x_1, \ldots, x_n$ have values satisfying φ. We define the induced subgraphs $H_1, \ldots, H_n$ and $H'_1, \ldots, H'_m$ of G'' by setting

- for every $i \in [n]$, $V(H_i) = \{x_i, y_i^1, y_i^2, y_i^3\}$ if $x_i = false$ and $V(H_i) = \{\overline{x}_i, y_i^1, y_i^2, y_i^3\}$, otherwise,
- for every $j \in [m]$, consider a literal $\ell = true$ in C_j and let $r \in \{u_j, v_j, w_j\}$ be the clause vertex adjacent to the variable vertex corresponding to ℓ, and then set $V(H'_j) = (\{r\} \cup Z_j)$ if $|C_j| = 3$ and $V(H'_j) = (\{r, w_j\} \cup Z_j)$ if $|C_j| = 2$.

Notice that $H_1, \ldots, H_n$ and $H'_1, \ldots, H'_m$ are copies of K_4, that is, they are in $\mathcal{H}$. Furthermore, all these subgraphs of G'' are disjoint, and distinct subgraphs have no adjacent edges. We set

$$X = V(G'') \setminus \Big(\big(\bigcup_{i=1}^{n} V(H_i) \big) \cup \big(\bigcup_{j=1}^{m} V(H'_j) \big) \Big).$$

Then $G'' - X$ is the disjoint union of $n+m$ copies of K_4 and we obtain that each connected component of $G'' - X$ is in $\mathcal{H}$. Because G' is planar, we have that the torso of X is planar as well. To see this, notice that the torso of X can be obtained from G' by edge contractions and replacements of Y-subgraphs, that is, subgraph isomorphic to the star $K_{1,3}$ whose central vertex has degree three in G', by triangles formed by the leaves of the star (see Figure 10.11). Since these operations preserve planarity, the torso of X is planar. Thus, G'' has a planar $\mathcal{H}$-modulator.

For the opposite direction, assume that G'' has a planar $\mathcal{H}$-modulator. Let $X \subseteq V(G'')$ be such that the torso of X is planar and $G'' - X$ is the disjoint union of copies of K_4. Consider $i \in [n]$. The clique $R_i = \{x_i, \overline{x}_i, y_i^1, y_i^2, y_i^3\}$ of size 5, so $X \cap R_i \neq \emptyset$. Observe that, by the construction of G'', any clique of size four containing a vertex of H_i is contained in R_i. Thus, $G'' - X$ has a connected component H_i that is copy of K_4 with $V(H_i) \subseteq R_i$. Therefore, at least one of the vertices x_i and $\overline{x}_i$ is in H_i. We set the value of the variable $x_i = true$ if $x_i \notin V(H_i)$ and $x_i = false$, otherwise. We perform this assignment of the values to all variables $x_1, \ldots, x_n$. We claim that this is a satisfying assignment for φ.

Consider $j \in [m]$. Suppose first that $|C_j| = 3$. Let $R'_j = \{u_j, v_j, w_j, z_j^1, z_j^2, z_j^3\}$. Since R'_j is a clique of size 6, $X \cap R'_j \neq \emptyset$. The construction of G'' implies that each clique of size four containing a vertex of R'_j is contained in R'_j. This means that $G'' - X$ has a connected component H'_j isomorphic to K_4 such that $V(H'_j) \subseteq R'_j$. Then at least one of the vertices u_j, v_j, w_j is in $V(H'_j)$. By symmetry, we can assume without loss of generality that $u_j \in V(H'_j)$. Also by symmetry, we can assume that u_j is adjacent to x_i for some $i \in [n]$. Because H_i and H'_j are disjoint and have no adjacent vertices, $x_i \notin V(H_i)$. This means that $x_i = true$ and the clause C_j is satisfied. The case $|C_j| = 2$ is analyzed in a similar way. Now we let $R'_j = \{u_j, v_j, w_j, z_j^1, z_j^2\}$ and have that R'_j is a clique of size 5. Then $X \cap R'_j \neq \emptyset$. We again have that each clique of size four containing a vertex of R'_j is contained in R'_j. Therefore, $G'' - X$ has a connected component H'_j isomorphic to K_4 such that $V(H'_j) \subseteq R'_j$. Recall that we assume that w_j is not adjacent to any variable vertex. Then either u_i or v_i is in $V(H'_j)$. We assume without loss of generality that $u_j \in V(H_j)$ and u_i are adjacent to x_i for some $i \in [n]$.

Because H_i and H'_j are disjoint and have no adjacent vertices, $x_i \notin V(H_i)$. Thus, $x_i = \mathit{true}$ and the clause C_j is satisfied. This proves that each clause is satisfied by our truth assignment for the variables. We conclude that φ has a satisfying assignment.

It is straightforward that G'' can be constructed in polynomial time. Therefore, $\mathcal{H}$-PLANARITY is NP-hard for $\mathcal{H} = \{K_4\}$. As the inclusion of $\mathcal{H}$-PLANARITY in NP is trivial, this concludes the proof. $\square$

In Theorem 2.6.2 and Theorem 2.6.3, the existence of the respective algorithms for $\mathsf{p} \in \{\mathsf{ptd}, \mathsf{ptw}\}$ is shown under condition that the VERTEX DELETION TO $\mathcal{H}$ problem parameterized by the solution size is FPT. While we leave open the question whether the existence of an FPT algorithm for checking that the $\mathcal{H}$-planar treedepth is at most k or the $\mathcal{H}$-planar treewidth at most k would imply that there is an FPT algorithm solving VERTEX DELETION TO $\mathcal{H}$, we note that for natural hereditary graph classes $\mathcal{H}$ for which VERTEX DELETION TO $\mathcal{H}$ is known to be W[1]-hard or W[2]-hard, deciding whether the elimination distance to $\mathcal{H}$ and whether the $\mathcal{H}$-planar treewidth at most k is also can be shown to be hard. This follows from the observation that for graphs of high vertex connectivity, the three problems are essentially equivalent. In particular, it can be noticed the following.

Observation 10.6.2. *Let G be a $4k+2$-connected graph for an integer $k \geq 0$. Then, for any class $\mathcal{H}$, a graph $G^* \in \mathcal{H}$ can be obtained by at most $4k$ vertex deletions from G if and only if the $\mathcal{H}$-planar treedepth of G is at most k.*

Proof. Suppose that there is a set $X \subseteq V(G)$ of size at most $4k$ such that $G - X \in \mathcal{H}$. Because $|X| \leq 4k$, $G - X$ is connected. Also, since $|X| \leq 4k$, there is a partition $\{X_1, \ldots, X_s\}$ of X such that $s \leq k$ and $|X_i| \leq 4$ for each $i \in [s]$. Notice that for each $i \in [s]$, $\mathsf{torso}(G_i, X_i)$, where $G_i = G - \bigcup_{j=1}^{i-1} X_j$ is planar. Then we decompose G by consecutively selecting $X_1, \ldots, X_s$.

For the opposite direction, we use induction on k. We prove that if the $\mathcal{H}$-planar treedepth of a graph G is at most k then a graph $G' \in \mathcal{H}$ can be obtained by deleting at most $4k$ vertices. The claim trivially holds if $G \in \mathcal{H}$ as the deletion distance to $\mathcal{H}$ is $0 \leq k$. In particular, this proves the claim for $k = 0$. Assume that $k \geq 1$ and $G \notin \mathcal{H}$. Because $\mathsf{ptd}_{\mathcal{H}}(G) \leq k$, there is a nonempty set $X \subseteq V(G)$ such that (i) for each connected component C of $G' = G - X$, $\mathsf{ptd}_{G'}(C) \leq k-1$ and (ii) $\mathsf{torso}(G, X_1)$ is planar. Because G is 6-connected, G is not planar and, therefore, $X \neq V(G)$. Consider an arbitrary connected component C of $G - S$. Because $\mathsf{torso}(G, X)$ is planar, $N_G(V(C)) \leq 4$. Since $N_G(V(C))$ cannot be a separator in a 6-connected graph, we obtain that $X = N_G(V(C))$ is of size at most four and $C = G'$ is a unique connected component of $G - X$. Because $|X| \leq 4$, G' is $4(k-1)+2$-connected and we can apply induction. Then there is a set $Y \subseteq V(G')$ such that $|Y| \leq 4(k-1)$ and $G^* = G' - Y \in \mathcal{H}$. Consider $Z = X \cup Y$. Then $G^* = G' - Y = G - Z$ and $|Z| \leq 4k$. This proves that the deletion distance of G to $\mathcal{H}$ is at most $4k$. $\square$

Given a computational lower bound for VERTEX DELETION TO $\mathcal{H}$, typically, it is easy to show that the hardness holds for highly connected graphs and for k divisible by four. Here, we give just one example. If was proved by Heggernes et al. [164] that PERFECT DELETION, that is, VERTEX DELETION TO $\mathcal{H}$ when $\mathcal{H}$ is the class of perfect graphs, is W[2]-hard when parameterized by the solution size. Then we can show the following theorem using Observation 10.6.2.

Theorem 10.6.3. *Let $\mathcal{H}$ be the class of perfect graphs. The problem of deciding whether the $\mathcal{H}$-planar treedepth is at most k is* W[2]*-hard when parameterized by k.*

Proof. We reduce from PERFECT DELETION. Let (G, k) be an instance of the problem. First, we can assume without loss of generality that $k = 4k'$ for an integer $k' \geq 0$. Otherwise, we add $4 - (k \bmod 4)$ disjoint copies of the cycle on five vertices to G using the fact that at least one vertex should be deleted form each odd hole to obtain a perfect graph. Then we observe that the class of perfect graphs is closed under adding universal vertices. This can be seen, for example, from the strong perfect graph theorem [63]. Let G' be the graph obtained from G by adding $6k'$ vertices, making them adjacent to each other, and adjacent to each every vertex of G. Then the deletion distance of G to perfect graphs is the same as the deletion distance of G'. Thus, the instance (G, k) of PERFECT DELETION is equivalent to the instance (G', k). By Observation 10.6.2, we obtain that $(G', 4k)$ is a yes-instance of PERFECT DELETION if and only if the $\mathcal{H}$-planar treedepth of G' is at most k'. This completes the proof. □

The same arguments also could be used for the case when $\mathcal{H}$ is the class of weakly chordal graph—it was proved by Heggernes et al. [164] that WEAKLY CHORDAL DELETION is W[2]-hard and the class is also closed under adding universal vertices. Also, it is known that the WHEEL-FREE DELETION, that is, the problem asking whether k vertices may be deleted to obtain a wheel-free graph (a graph is wheel-free if it does not contain a *wheel*, i.e., a graph obtained from a cycle by adding a universal vertex, as an induced subgraph) is W[2]-hard when parameterized by k by the result of Lokshtanov [223]. We remark that it is possible to show that the lower bound holds for highly connected graphs and obtain the W[2]-hardness for the $\mathcal{H}$-planar treedepth when $\mathcal{H}$ is the class of wheel-free graphs.

Finally in this section, we note that the variant of Observation 10.6.2 holds for $\mathcal{H}$-planar treewidth.

Observation 10.6.4. *Let G be a $(\max\{4, k\} + 2)$-connected graph for an integer $k \geq 0$. Then, for any class $\mathcal{H}$, a graph $G^* \in \mathcal{H}$ can be obtained by at most k vertex deletions from G if and only if the $\mathcal{H}$-planar treewidth of G is at most k.*

Then the lower bound for DELETION TO $\mathcal{H}$ from [164, 223] can be used to show the W[2]-hardness for deciding whether the $\mathcal{H}$-planar treewidth is at most k.

Part V

Conclusion and research directions

CHAPTER 11

Concluding remarks

In this final chapter, we discuss several consequences and open questions that emerge from our results.

11.1 Perspectives on our structure theorem

In Chapter 5, we prove a decomposition-based min-max theorem for the structure of graphs excluding as minors edge-apex graphs or, alternatively, graphs embedded in the pinched sphere. We prove that such graphs can be tree-decomposed so that each torso contains a set of vertices of bounded bidimensionality that can be identified to obtain a graph embeddable in the projective plane. Moreover, this decomposition optimally determines the structure of the excluded edge-apex graphs. If the excluded graph is embeddable in a surface, analogous decomposition theorems are proved in [302], for every surface. In that sense this work can be seen as the first step to extend the results of [302] to pseudosurfaces, in particular to those that, according to Knor [199], define graphs classes that are minor-closed. A first question is hence the following.

Question 1. *For each surface Σ, what is the structure of graphs excluding as a minor a graph embeddable in the pinched version Σ° of Σ?*

In this direction we believe that the decomposition theorems for pinched surfaces of higher genus are not expected to avoid the presence of apices.

Notice that the operations that are applied to the torsos of our decomposition are not removals of low bidimensionality vertex sets, as it is the case of the interpretation of the GMST in [302], but vertex identifications of them. In our structural theorem, given the absence of apices, they serve in order to shrink the "few" vortices to vertices that will give rise to an embedding in the projective plane or in the sphere. In fact, we may combine vertex removals and vertex identifications in order to describe the modification operations for the version of the GMST proposed in [302]: when it comes to apices we remove vertex sets of small size, and when it comes to vortices we identify them as vertex sets of low bidimensionality in order to obtain some surface embedding of the resulting graph. In that sense, *vertex removal* of bounded size is the correct modification operation for *apices* and *vertex identification* of small bidimensionality is the correct modification operation for *vortices*.

We obtain the first structure theorem with vortices but no apices. Here, the surface is either the sphere (Theorem 5.2.37) or the projective plane (Theorem 5.2.36). A natural follow-up question is thus the following.

Question 2. *For each surface Σ, does there exist a graph class $\mathcal{H}_\Sigma$ such that:*

- *for each graph $H \in \mathcal{H}_\Sigma$, there exists a constant c_H depending on $|V(H)|$ such that each H-minor-free graph G can be tree-decomposed so that each torso has a c_H-almost embedding in Σ, and*
- *for each $h \in \mathbb{N}$, there exists a graph $H \in \mathcal{H}_\Sigma$ that cannot be tree-decomposed so that each torso has an h-almost embedding in Σ?*

Let us mention that Thilikos and Wiederrecht introduced in [303] the concept of the vga-hierarchy where v stands for "vortices", g for "genus", and a for "apices". For each $\mathsf{x} \subseteq \{\mathsf{v}, \mathsf{g}, \mathsf{a}\}$, we can try to find an infinite class of graphs $\mathcal{H}_\mathsf{x}$ such that a graph excluding $H \in \mathcal{H}$ as a minor can be tree-decomposed so that each torso has an almost embedding (or an embedding if $\mathsf{v} \notin \mathsf{x}$) in a surface of bounded genus (or the sphere if $\mathsf{g} \notin \mathsf{x}$) after removing a bounded number of apices (or none if $\mathsf{a} \notin \mathsf{x}$). $\mathcal{H}_\mathsf{vga}$ is of course the class of all graphs, as proved in the GMST. Thilikos and Wiederrecht settle in [303] the case of $\mathcal{H}_\mathsf{ga}$. In Theorem 5.2.37, we settle the case of $\mathcal{H}_\mathsf{v}$, while Question 2 corresponds to the case of $\mathcal{H}_\mathsf{vg}$.

Another direction of research is whether one may use the decomposition of Theorem 2.1.2 (or Theorem 2.1.4) for algorithmic purposes.

Question 3. *Are there interesting problems for which the presence of $\mathscr{J}_k$ (or of $\mathscr{J}_k$ and $\mathscr{C}_k$) as a minor in the input graph directly certifies an answer?*

In other words, we search for problems that behave well with respect to vortices, but not with respect to apices.

A last question for this chapter is the following.

Question 4. *Can the functional dependencies of our results (i.e., the function f mapping H to c_H in Theorem 2.1.2 or Theorem 2.1.4) be reduced to polynomial ones?*

The only obstacle for this resides in the already exponential dependencies of the GMST in [195] that is the departure point of our proof. Apart from this, all bounds generated by our proofs are polynomial. In the light of the result of Gorsky, Seweryn, and Wiederrecht that just appeared in [148] reducing the dependencies of the GMST to a polynomial one, the functional dependencies of our results are likely to be reducible as well.

11.2 Open problems on identifications

In Chapter 6, we initiate the study of graph modification problems where the modification operation is vertex identification. We defined the problem IDENTIFICATION TO $\mathcal{H}$ and studied the case where the target class $\mathcal{H}$ is the class of forests, denoted by $\mathcal{F}$. We prove that IDENTIFICATION TO $\mathcal{F}$ is NP-hard, and provide a linear kernel as well as an FPT-algorithm for the problem that are derived from similar results for VERTEX COVER. Any improvement on the parameterized complexity of VERTEX COVER would immediately imply an improvement on our results. Additionally, we prove in Theorem 6.2.13 that the obstructions of the set $\mathcal{F}^{(k)}$ of yes-instances of k-IDENTIFICATION TO $\mathcal{F}$ have at most $2k + 4$ vertices. We actually conjecture the following.

Conjecture 1. *For each $k \in \mathbb{N}_{\geq 3}$, the obstruction of maximum size of $\mathcal{F}^{(k)}$ is C_{2k+1}.*

Dinneen and Lai proved in [92] that C_{2k+1} is the maximum connected obstruction of the set $\mathcal{V}_k$ of yes-instances of k-Vertex Cover. Given that a disconnected obstruction of $\mathcal{V}_k$ is the disjoint union of obstructions of lower levels, we can easily prove that the only obstruction of $\mathcal{V}_k$ of larger size that C_{2k+1} is the matching $(k+1) \cdot K_2$. To prove Conjecture 1, we can deduce from the above and from the proof of Theorem 6.2.13 that it would be sufficient to prove that (a) no obstruction of $\mathcal{F}^{(k)}$ can be obtained from $(k+1) \cdot K_2$ by adding edges and (b) the marguerite $(k+1) * K_3$ (for even k) is the only obstruction of $\mathcal{F}^{(k)}$ that is also an obstruction of $\mathcal{F}^{(k+1)}$.

The universal obstructions that we obtain for Identification to $\mathcal{F}$ are packings of triangles, marguerites, and cycles (Theorem 6.3.1). Going further, we can wonder about the universal obstructions of Identification to Minor-closedness.

Question 5. *For each minor-closed graph class $\mathcal{H}$, what are the universal obstructions of* Identification to $\mathcal{H}$*?*

We conjecture that they are, for each $H \in \mathsf{obs}(\mathcal{H})$:

- a packing of H,
- for each $v \in V(H)$, copies of H identified together at vertex v, and
- for each $e = uv \in E(H)$, copies of $H - e$ glued in a cyclic manner so that the vertex v in copy i is identified to vertex u in copy $i+1$.

As we introduced in Section 6.5, we may consider universal obstructions in terms of identification minors instead of minors, which reduce to a unique universal obstruction, that is the marguerite of triangles. We can again wonder about the universal obstructions of Identification to Minor-closedness, this time for identification minors.

Question 6. *For each minor-closed graph class $\mathcal{H}$, what are the universal obstructions of* Identification to $\mathcal{H}$ *in terms of identification minors?*

Here, we conjecture that they are, for each $H \in \mathsf{obs}(\mathcal{H})$ and for each $v \in V(H)$, copies of H identified together at vertex v.

For each graph class $\mathcal{H}$, recall that $\mathsf{ec}_{\mathcal{H}}$ is the parameter mapping a graph G to the minimum number of edge contractions that can transform G to a graph in $\mathcal{H}$, and let $\mathsf{id}_{\mathcal{H}}$ be the graph parameter mapping a graph G to the minimum k such that (G, k) is a yes-instance of Identification to $\mathcal{H}$. We prove in Lemma 6.4.1 that $\mathsf{ec}_{\mathcal{F}} \sim \mathsf{id}_{\mathcal{F}}$. While it is easy to see that $\mathsf{id}_{\mathcal{H}}(G) \leq 2 \cdot \mathsf{ec}_{\mathcal{H}}(G)$, we also conjecture that an upper bound as the one of Lemma 6.4.1 holds for every minor-closed class $\mathcal{H}$.

Conjecture 2. *For every minor-closed graph class $\mathcal{H}$, there is a function $f_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ such that, for every G, $\mathsf{ec}_{\mathcal{H}} \leq f_{\mathcal{H}}(\mathsf{id}_{\mathcal{H}}(G))$.*

Note that Contraction to $\mathcal{H}$ is known to be W[1]-hard, parameterized by the solution size, for several families $\mathcal{H}$ that are *not* minor-closed, such as chordal graph or split graphs (see [9] and the references cited therein). However, when $\mathcal{H}$ is minor-closed, the recent meta-algorithmic results in [120] (further generalized in [287]) imply that Contraction to $\mathcal{H}$ is (constructively) FPT (see [165, 209] for explicit algorithms for some particular families). Also, as it has been proved in [165], Contraction to Forest is not expected to admit a polynomial kernel. Interestingly, the kernelization we give in this chapter for Identification to Forest, under the light of the polynomial-gap functional equivalence of Lemma 6.4.1, can be seen as some kind of "functional kernel" for Contraction to Forest.

11.3 Concluding notes on bounded size modulators

For a large family of graph modification problems involving a bounded number of vertices, if the target class $\mathcal{H}$ is minor-closed, we provide in Chapter 7 an algorithm solving the problem in time $2^{\mathsf{poly}(k)} \cdot n^2$. This is actually the same running time as the best known running time for VERTEX DELETION TO $\mathcal{H}$ [235] (up to an extra additive constant of one in the degree of the polynomial function $\mathsf{poly}(k)$ that is absolutely negligible compared to the total degree that depends wildly on the size of the obstructions of $\mathcal{H}$). For the other graph modification problems encompassed by our result, such as EDGE DELETION TO $\mathcal{H}$, EDGE CONTRACTION TO $\mathcal{H}$, VERTEX IDENTIFICATION TO $\mathcal{H}$, or INDEPENDENT SET DELETION TO $\mathcal{H}$, the only minor-closed $\mathcal{H}$ for which an algorithm with an explicit parametric dependence on the solution size was known, to our knowledge, were the classes of forests and of union of paths [165, 218, 219, 307]. Other problems, such as MATCHING DELETION TO $\mathcal{H}$, MATCHING CONTRACTION TO $\mathcal{H}$, INDUCED STAR DELETION TO $\mathcal{H}$, or SUBGRAPH COMPLEMENTATION TO $\mathcal{H}$, were not even considered yet from the parameterized complexity viewpoint, other than in the meta-theorem of [287]. A natural question is the following.

Question 7. *Can we find a faster algorithm for $\mathcal{L}$-REPLACEMENT TO $\mathcal{H}$ when $\mathcal{L}$ is hereditary and $\mathcal{H}$ is a minor-closed graph class, and, as a first step, for* VERTEX DELETION TO $\mathcal{H}$*?*

Concerning the parametric dependence, the question is more particularly whether $\mathsf{poly}(k)$ could be replaced by $c \cdot k^d$ for some constant c depending on $\mathcal{H}$ and some *universal* constant d (independent of $\mathcal{H}$). This dependence on $\mathcal{H}$ comes from the use of the irrelevant vertex vertex technique of [271]. Thus, improving more the parametric dependence would certainly require coming up with new techniques. On the other hand, given the recent results of [205] for minor containment, it is worth studying whether the quadratic dependence on n could be improved to an almost-linear dependence while maintaining a good dependence on k. Note that the approach of [205] heavily uses Courcelle's theorem [67], which would require to be translated to a plausibly very involved dynamic programming algorithm to keep a good parametric dependence on k.

On the other hand, we are not aware of any lower bound, assuming the Exponential Time Hypothesis [169], stronger than $2^{o(k)} \cdot n^{\mathcal{O}(1)}$, which follows quite easily from known results for VERTEX COVER. Proving stronger lower bounds seems to be quite challenging.

In the bounded genus case, we reduce the running time to $2^{\mathcal{O}(k^9)} \cdot n^2$ thanks to some improvement on the irrelevant vertex technique. To our knowledge, this is the first bounded genus result with an explicit parametric dependence on the solution size for the other graph modification problems encompassed by our result. This does not match the parametric dependence on the running time of $2^{\mathcal{O}(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$ for VERTEX DELETION TO $\mathcal{H}$ [202] for $\mathcal{H}$ of bounded genus, though we possibly have a better dependence on n. Hence, we can ask the following.

Question 8. *Can we solve $\mathcal{L}$-REPLACEMENT TO $\mathcal{H}$ as fast as* VERTEX DELETION TO $\mathcal{H}$ *(in terms of parametric dependence) when $\mathcal{L}$ is hereditary and $\mathcal{H}$ is class of graphs embeddable in a surface of bounded genus?*

Given that we require the replacement action $\mathcal{L}$ to be hereditary for our irrelevant vertex technique to work, unfortunately we restrict the graph modification problems that we solve. For instance, PLANAR SUBGRAPH ISOMORPHISM can be expressed as an $\mathcal{L}$-REPLACEMENT TO PLANAR problem for a specific $\mathcal{L}$, which is not hereditary. Hence, we do not encompass this problem in our general algorithm, while such an algorithm is provided in [121], where the constraint about $\mathcal{L}$ being hereditary is not required. While most of the "reasonable" modification problems correspond to a hereditary replacement action, it is worth investigating whether our result can be extended to non-hereditary replacement actions. In other words:

Question 9. *Can we drop the hereditary condition for* $\mathcal{L}$-Replacement to $\mathcal{H}$ *and still solve the problem within the same running time?*

11.4 Beyond elimination distance

For a minor-closed graph class $\mathcal{H}$, we prove in Chapter 8 that Vertex Deletion to $\mathcal{H}$ can be solved in time $2^{\mathsf{poly}(k)} \cdot n^2$ and that Elimination Distance to $\mathcal{H}$ can be solved in time $2^{2^{2^{\mathsf{poly}(k)}}} \cdot n^2$, and in time $2^{2^{c \cdot k^2 \log k}} \cdot n^2$ and $2^{\mathsf{poly}(k)} \cdot n^3$ in the case where the obstruction set of $\mathcal{H}$ contains an apex-graph. As in Question 7, a natural question is whether a faster algorithm exists.

Question 10. *Can we find a faster algorithm for* Elimination Distance to $\mathcal{H}$ *when* $\mathcal{H}$ *is a minor-closed graph class?*

Given that the degree of poly and c heavily depend again on the size of the obstructions of $\mathcal{H}$ due to the use the irrelevant vertex technique, there is not much hope to improve those before answering positively Question 7 (more particularly for Vertex Deletion to $\mathcal{H}$). Similarly, improving the dependence on n for Vertex Deletion to $\mathcal{H}$ would give an idea of how to improve the dependence on n for Elimination Distance to $\mathcal{H}$.

Another way to answer Question 10 positively would be to drop the running time of Elimination Distance to $\mathcal{H}$ to $2^{\mathsf{poly}(k)} \cdot n^2$ for *every* minor-closed graph class $\mathcal{H}$. We tend to believe that this should be possible. However, it seems to require to use branching ingeniously and, in particular, to find equivalent instances of Elimination Distance to $\mathcal{H}$ with a decreasing value of k.

We also proposed an XP-algorithm for Elimination Distance to $\mathcal{H}$ parameterized by the treewidth of the input graph (with running time $n^{\mathcal{O}(\mathsf{tw}^2)}$). As mentioned in [6], the existence of an FPT-algorithm for Elimination Distance to $\mathcal{H}$, parameterized by treewidth, remains wide open and this is the case even in the very special case where $\mathcal{H}$ contains only the empty graph, where Elimination Distance to $\mathcal{H}$ is equivalent to the problem of computing treedepth. The first question in this direction is hence:

Question 11. *Can we compute the treedepth of a graph parameterized by its treewidth in* FPT*-time?*

Another direction of research would be to improve the bounds on the size of the obstructions given in Theorem 2.4.4. Let us thus ask the following.

Question 12. *What is the maximum size of an obstruction of the set* $\mathcal{E}_k(\mathcal{H})$ *of* yes*-instances of* k-Elimination Distance to $\mathcal{H}$ *when* $\mathcal{H}$ *is minor-closed?*

We believe again that any substantial improvement should demand novel methodologies that go beyond the irrelevant vertex technique.

As mentioned in Section 1.5, Elimination Distance to $\mathcal{H}$ asks, given an instance (G, k), whether $\mathcal{H}\text{-}\mathsf{td}(G) \leq k$. A natural question is thus whether the methods used to solve the problem for $\mathcal{H}$-td can be generalized to any parameter $\mathcal{H}$-p. That is:

Question 13. *For any graph parameter* p *and any (apex-)minor-closed graph class* $\mathcal{H}$*, is there an algorithm checking whether* $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ *as fast as the one for* $\mathsf{p} = \mathsf{td}$*?*

We believe that this is true for any minor-monotone parameter p such that $\mathsf{tw} \preceq \mathsf{p} \preceq \mathsf{size}$. The irrelevant vertex technique (Proposition 8.2.4) works similarly in this case. The only missing ingredient is a dynamic programming algorithm solving the problem on graphs of bounded treewidth. A first step in this direction would be to solve the problem for $\mathsf{p} = \mathsf{tw}$, that is, to compute the

$\mathcal{H}$-treewidth of a graph parameterized by tw, and then by $\mathcal{H}$-tw, with the same running time as Elimination Distance to $\mathcal{H}$. For the parameterization by tw, one may consider combining a known dynamic programming algorithm for computing the treewidth [33, 204] with the representative-based technique of [24].

Going even further, instead of vertex deletions, we could consider other modifications. That is, we could consider $\mathcal{L}$-Replacement to $\mathcal{H}$, not parameterized by the size of the modulator, but by another parameter. Again, the techniques are likely to transfer there, the main obstacle being to design a dynamic programming algorithm on bounded treewidth graphs.

11.5 Towards odd-minor-closedness

In Chapter 9, we study the complexity of several problems parameterized by bipartite treewidth, denoted by btw. In particular, our results extend the graph classes for which Vertex Cover/ Independent Set, Maximum Weighted Cut, Odd Cycle Transversal, and Maximum Weighted Cut are polynomial-time solvable. A number of interesting questions remain open.

We are still far from a full classification of the variants that are para-NP-complete, and those that are not (FPT or XP). For instance, concerning H-Subgraph-Cover, we provided FPT-algorithms when H is a clique (Corollary 9.5.6). This case is particularly well-behaved because we know that in a tree decomposition every clique appears in a bag. On the other hand, as an immediate consequence of the result of Yannakakis [314] (Proposition 9.5.28), we know that H-Subgraph-Cover is para-NP-complete for every bipartite graph H containing P_3 (cf. Subsection 9.5.5). We do not know what happens when H is non-bipartite and is not a clique. An apparently simple but challenging case is C_5-Subgraph-Cover (or any other larger odd cycle).

Question 14. *Is C_5-Subgraph-Cover parameterized by* btw *solvable in* FPT*-time?*

The main difficulty seems to be that C_5-Subgraph-Cover does not have the gluing property, which is the main ingredient in our proofs to show that a problem is nice, and therefore to obtain an FPT-algorithm. We do not exclude the possibility that the problem is para-NP-complete, as we were not able to obtain even an XP algorithm.

Most of our para-NP-completeness results consist in proving NP-completeness on bipartite graphs (i.e., those with bipartite treewidth zero). There are two exceptions, which can be found in [171]. On the one hand, the NP-completeness of 3-Coloring on graphs with odd cycle transversal at most three and on the other hand, the NP-completeness of H-Scattered-Packing parameterized by q-$\mathcal{B}$-treewidth for every integer $q \geq 2$. None of our hardness results really exploits the structure of bipartite tree decompositions (i.e., for $q = 1$), beyond being bipartite or having bounded odd cycle transversal. Hence, we ask the following.

Question 15. *Is there a problem whose complexity differs when parameterized by* oct *or by* btw*?*

Going further away, in the light of the results obtained in Chapter 7 for Vertex Deletion to $\mathcal{H}$ (and more generally $\mathcal{L}$-R-$\mathcal{H}$), we can ask the following.

Question 16. *Given an odd-minor-closed graph class $\mathcal{H}$, can we solve* Vertex Deletion to $\mathcal{H}$ *in* FPT*-time parameterized by the solution size as fast as when the target graph class is minor-closed?*

There are two main obstacles towards such an algorithm, if we want to use techniques similar to the ones of Chapter 7. The first one concerns the flat wall theorem (Subsection 3.1.2): one of its outputs is a big clique that is a minor of G. When $\mathcal{H}$ is minor-closed, we can trivially output a no-answer. This is not the case for an odd-minor-closed graph class $\mathcal{H}$, given that an (odd-minor-)obstruction of $\mathcal{H}$ might not be found in a big clique-minor. An easy way out is to assume that one of the (odd-minor-) obstructions of $\mathcal{H}$ is bipartite. In this case, we can actually find a packing of this obstruction in a big clique-minor, by applying results from [137]. Otherwise, we may adapt techniques from [126] to find an irrelevant vertex inside a big clique-minor. Unfortunately, this irrelevant vertex technique has a huge parametric dependence. Therefore, this would not be much help towards answering Question 16. The second obstacle concerns the irrelevant vertex technique, which has no parity conditions, something crucial when working with odd-minors. In the extended abstract of [193], Kawarabayashi, Reed, and Wollan hint towards results from Schrijver [288, 289] for the creation of an irrelevant vertex technique with parity conditions. This could be a promising direction towards solving Vertex Deletion to Odd-minor-closedness.

11.6 Further research on unbounded bidimensionality modulators

In Chapter 10, we develop a new irrelevant vertex technique to prove that $\mathcal{H}$-planarity is solvable in polynomial time under some mild conditions on $\mathcal{H}$ (cf. Theorem 2.6.1). We proceed with a few remarks on the conditions on $\mathcal{H}$.

First, it is important that we demand the torso of the modulator to be planar rather than to allow the whole modulator to be planar. Without this condition, the problem becomes NP-hard. Indeed, Farrugia [109] proved that for two graph classes $\mathcal{P}$ and $\mathcal{Q}$ that are hereditary and closed under disjoint union, the problem of deciding whether a graph G admits a partition (A, B) of $V(G)$ such that $G[A] \in \mathcal{P}$ and $G[B] \in \mathcal{Q}$ is NP-hard, unless $\mathcal{P}$ and $\mathcal{Q}$ are both the class of edgeless graphs. Hence, for any hereditary graph class $\mathcal{H}$ closed under the disjoint union operation, the problem of deciding, given a graph G, whether there exists $S \subseteq V(G)$ such that $G[S]$ is planar and that, for each $C \in \mathsf{cc}(G - S)$, $C \in \mathcal{H}$, is NP-hard.

Second, the necessity of $\mathcal{H}$ to be a polynomial-time decidable graph class is apparent—it is easy to show that NP-hardness of deciding whether a graph is in $\mathcal{H}$ implies NP-hardness of $\mathcal{H}$-Planarity as well. However, the necessity of $\mathcal{H}$ to be a hereditary property is less obvious. We prove in Theorem 10.6.1 that for non-hereditary classes $\mathcal{H}$, $\mathcal{H}$-Planarity becomes NP-hard.

Finally, we do not know whether CMSO-definability of $\mathcal{H}$ is necessary. The CMSO-definability condition is needed because we are using the meta-theorem of Lokshtanov, Ramanujan, Saurabh, and Zehavi from [224]. This meta-theorem is a powerful tool allowing us, via unbreakable graphs, to reduce solving $\mathcal{H}$-Planarity to solving $\mathcal{H}^{(k)}$-Planarity for bounded k, making it possible to apply our version of the irrelevant vertex technique. We thus ask the following.

Question 17. *Is there a graph class $\mathcal{H}$ that is hereditary and polynomial-time decidable, but not CMSO-definable, such that $\mathcal{H}$-Planarity is NP-hard?*

Using the meta-theorem of [224] also comes with the following drawback: Theorem 2.6.1 is non-constructive. Similar to the results in [224], it allows us to infer the existence of a polynomial-time algorithm for every CMSO formula φ, but it does not provide the actual algorithm. The reason for that is that [224] relies on the existence of representative subgraphs for equivalence classes of bounded-boundary graphs but does not provide a procedure to compute such representatives. In other words, Theorem 2.6.1 provides a *non-constructive* polynomial-time algorithm. Our next question is thus:

Question 18. *Is there a constructive polynomial-time algorithm solving $\mathcal{H}$-Planarity for any graph class $\mathcal{H}$ that is hereditary, CMSO-definable, and polynomial-time decidable?*

A last drawback of the meta-theorem of [224] is the following: Theorem 2.6.2 and Theorem 2.6.3 are not *uniform in k*. That is, for each $k \in \mathbb{N}$, they output a different algorithm checking whether $\mathcal{H}\text{-}\mathsf{ptd}(G) \leq k$ or $\mathcal{H}\text{-}\mathsf{ptw}(G) \leq k$. This comes from the fact that the constants on the unbreakable graphs we require depend on k. Hence the following question.

Question 19. *Is there an* FPT*-algorithm that, given a graph G and $k \in \mathbb{N}$, decides, uniformly in k, whether $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$, for $\mathsf{p} \in \{\mathsf{ptd}, \mathsf{ptw}\}$, under the same conditions for $\mathcal{H}$ as in Theorem 2.6.2 and Theorem 2.6.3?*

Other tools than the meta-theorem of [224] can perhaps be used in order to answer Question 17, Question 18, and Question 19. In particular, it can be noted that we aim to decompose in a special way the input graph via separators of small size, and graph decompositions of this type were introduced by Grohe [150]. It is very interesting whether avoiding using the meta-theorem is possible.

All our techniques heavily rely on the planarity of the torso of X. In particular, this concerns our method for finding an irrelevant vertex inside a flat wall no matter how the graph outside the modulator interacts with this flat wall. For this, we present our techniques for the $\mathcal{H}$-Planarity problem and later, in Section 10.3 and Section 10.4 we explain how these techniques can be extended on graphs where $\mathcal{H}$-ptd or $\mathcal{H}$-ptw is bounded. However, ptd and ptw are not the only minor-monotone parameters to consider, further than td and tw. We actually conjecture the following.

Conjecture 3. *Let $\mathcal{H}$ be a hereditary and CMSO-definable graph class and* p *be a minor-monotone graph parameter. Then an* FPT*-algorithm for checking $\mathcal{H}\text{-}\mathsf{size}(G) \leq k$ implies an* FPT*-algorithm for checking $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$.*

We believe that our techniques are a promising departure point for resolving this conjecture. However, this appears to be a quite challenging task.

On the other hand, Agrawal, Kanesh, Lokshtanov, Panolan, Ramanujan, Saurabh, and Zehavi proved in [6], that, for $\mathsf{p} \in \{\mathsf{td}, \mathsf{tw}\}$, an FPT-algorithm for checking $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ implies an FPT-algorithm for checking $\mathcal{H}\text{-}\mathsf{size}(G) \leq k$. We can thus wonder whether this result holds for any minor-monotone parameter p. We actually believe that this holds for $\mathsf{tw} \leq \mathsf{p} \leq \mathsf{size}$. More generally, we conjecture the following.

Conjecture 4. *Let $\mathcal{H}$ be a hereditary and CMSO-definable graph class and p, p' be two minor-monotone graph parameters such that $\mathsf{tw} \leq \mathsf{p}, \mathsf{p}' \leq \mathsf{size}$. Then an* FPT*-algorithm for checking $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ implies an* FPT*-algorithm for checking $\mathcal{H}\text{-}\mathsf{p}'(G) \leq k$.*

On the other hand, we conjecture that this does not hold if $\mathsf{hw} \leq \mathsf{p} \leq \mathsf{tw}$.

Conjecture 5. *There exist a graph class $\mathcal{H}$ that is hereditary, CMSO-definable and polynomial-time decidable and a graph parameter* p *that is minor-monotone such that checking whether $\mathcal{H}\text{-}\mathsf{p}(G) \leq k$ can be done in* FPT*-time, but checking whether $\mathcal{H}\text{-}\mathsf{size}(G) \leq k$ is* W[1]*-hard.*

The intuition is that there could be a subclass of planar graphs $\mathcal{H}$ such that checking whether $\mathcal{H}\text{-}\mathsf{ptd}(G) \leq k$ reduces to checking whether $\mathsf{ptd}(G) \leq k + 1$, which is solvable in FPT-time by Theorem 2.6.2. And it is unlikely that, for every subclass of planar graphs, Vertex Deletion to $\mathcal{H}$ is in FPT.

Regarding applications, we stress that Theorem 10.5.6, Theorem 10.5.9, Theorem 10.5.14, and their corollaries for graphs with bounded $\mathcal{H}$-ptd and $\mathcal{H}$-ptw are only indicative snapshots of the algorithmic applicability of $\mathcal{H}$-planarity, $\mathcal{H}$-planar treewidth, and $\mathcal{H}$-planar treedepth. Numerous planar applications exist in various algorithmic subfields, ranging from distributed algorithms to kernelization and subexponential algorithms. Of course, not all such methods and results for planar graphs could be transferred even to $\mathcal{H}$-planar graphs. Exploring the full set of algorithmic applications of $\mathcal{H}$-planarity, $\mathcal{H}$-planar treewidth, and $\mathcal{H}$-planar treedepth escapes the purposes of this thesis. However, we expect that our results will appear to be useful in extending algorithmic paradigms where dynamic programming on graphs of bounded treewidth can be extended to dynamic programming on graphs of bounded $\mathcal{H}$-treewidth.

Bibliography

[1] Karl R. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness IV: on completeness for W[P] and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995. doi:10.1016/0168-0072(94)00034-Z. 5

[2] Isolde Adler, Frederic Dorn, Fedor V. Fomin, Ignasi Sau, and Dimitrios M. Thilikos. Faster parameterized algorithms for minor containment. *Theoretical Computer Science*, 412(50):7018–7028, 2011. doi:10.1016/j.tcs.2011.09.015. 54

[3] Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 641–650, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347153. 17, 46

[4] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Irrelevant vertices for the planar disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 122:815–843, 2017. doi:10.1016/J.JCTB.2016.10.001. 46, 68

[5] Akanksha Agrawal, Pallavi Jain, Lawqueen Kanesh, Pranabendu Misra, and Saket Saurabh. Exploring the kernelization borders for hitting cycles. In *Proc. of the 13th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 115 of *LIPIcs*, pages 14:1–14:14, 2018. doi:10.4230/LIPICS.IPEC.2018.14. 130

[6] Akanksha Agrawal, Lawqueen Kanesh, Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Deleting, Eliminating and Decomposing to Hereditary Classes Are All FPT-Equivalent. In *Proc. of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1976–2004, 2022. doi:10.1137/1.9781611977073.79. x, 14, 22, 23, 39, 265, 277, 280

[7] Akanksha Agrawal, Lawqueen Kanesh, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. An FPT algorithm for elimination distance to bounded degree graphs. In *Proc. of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 187 of *LIPIcs*, pages 5:1–5:11, 2021. doi:10.4230/LIPIcs.STACS.2021.5. 14, 22

[8] Akanksha Agrawal, Lawqueen Kanesh, Saket Saurabh, and Prafullkumar Tale. Paths to trees and cacti. *Theoretical Computer Science*, 860:98–116, 2021. doi:10.1016/J.TCS.2021.01.033. 129

[9] Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Split contraction: The untold story. *ACM Transactions on Computational Theory*, 11(3):18:1–18:22, 2019. doi:10.1145/3319909. 275

[10] Akanksha Agrawal and M. S. Ramanujan. On the parameterized complexity of clique elimination distance. In *Proc. of the 15th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 180 of *LIPIcs*, pages 1:1–1:13, 2020. doi:10.4230/LIPIcs.IPEC.2020.1. 22

[11] Akanksha Agrawal, Saket Saurabh, and Prafullkumar Tale. On the parameterized complexity of contraction to generalization of trees. *Theory of Computing Systems*, 63(3):587–614, 2019. doi:10.1007/S00224-018-9892-Z. 129

[12] Ernst Althaus and Sarah Ziegler. Optimal tree decompositions revisited: A simpler linear-time fpt algorithm. In *Graphs and Combinatorial Optimization: from Theory to Applications*, volume 5 of *AIRO Springer Series*, pages 67–78. Springer, 2021. doi:10.1007/978-3-030-63072-0_6. 54

[13] Eyal Amir. Approximation algorithms for treewidth. *Algorithmica*, 56(4):448–479, 2010. doi:10.1007/S00453-008-9180-4. 8

[14] Dhanyamol Antony, Sagartanu Pal, and R. B. Sandeep. Algorithms for subgraph complementation to some classes of graphs. *Information Processing Letters*, 188:106530, 2025. doi:10.1016/J.IPL.2024.106530. 131

[15] Kenneth Appel and Wolfgang Haken. *Every planar map is four colorable.* American Mathematical Society, 1989. 262

[16] N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. On polynomial kernelization of h-free edge deletion. *Algorithmica*, 79(3):654–666, 2017. doi:10.1007/S00453-016-0215-Y. 14

[17] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987. doi:10.1137/0608024. 8, 165

[18] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991. doi:10.1016/0196-6774(91)90006-K. ix, 8, 24, 43, 47, 54, 263

[19] Arash Asadi, Luke Postle, and Robin Thomas. Minor-minimal non-projective planar graphs with an internal 3-separation. *Electronic Notes Discrete Mathematics*, 38:69–75, 2011. doi:10.1016/J.ENDM.2011.09.012. 106

[20] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM (JACM)*, 41(1):153–180, 1994. doi:10.1145/174644.174650. 265

[21] R. Balasubramanian, Michael R. Fellows, and Venkatesh Raman. An improved fixed-parameter algorithm for vertex cover. *Information Processing Letters*, 65(3):163–168, 1998. doi:10.1016/S0020-0190(97)00213-5. 7

[22] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Optimal algorithms for hitting (topological) minors on graphs of bounded treewidth. In *Proc. of the 12th International Symposium on*

Parameterized and Exact Computation (IPEC), volume 89 of *LIPIcs*, pages 4:1–4:12, 2017. doi:10.4230/LIPIcs.IPEC.2017.4. 14

[23] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms. *Theoretical Computer Science*, 814:135–152, 2020. doi:10.1016/j.tcs.2020.01.026. 47, 54, 55, 159

[24] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. IV. An optimal algorithm. *SIAM Journal on Computing*, 52(4):865–912, 2023. doi:10.1137/21M140482X. 46, 55, 62, 68, 132, 145, 152, 175, 278

[25] Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *Journal of Artificial Intelligence Research*, 12:219–234, 2000. doi:10.1613/JAIR.638. 11

[26] Mahdi Belbasi and Martin Fürer. Finding all leftmost separators of size $\le k$. In *Proc. of the 15th International Conference on Combinatorial Optimization and Applications (COCOA)*, volume 13135 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2021. doi:10.1007/978-3-030-92681-6_23. 8

[27] Mahdi Belbasi and Martin Fürer. An improvement of reed's treewidth approximation. *Journal of Graph Algorithms and Applications*, 26(2):257–282, 2022. doi:10.7155/JGAA.00593. 8

[28] Patrick Bellenbaum and Reinhard Diestel. Two short proofs concerning tree-decompositions. *Combinatorics, Probability and Computing*, 11(6):541–547, 2002. doi:10.1017/S0963548302005369. 190

[29] Tatiana Belova and Ivan Bliznets. Hardness of approximation for h-free edge modification problems: Towards a dichotomy. In *Proc. of the 33rd International Symposium on Algorithms and Computation (ISAAC)*, volume 248 of *LIPIcs*, pages 24:1–24:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ISAAC.2022.24. 10

[30] Ivan Bliznets, Marek Cygan, Pawel Komosa, and Michal Pilipczuk. Hardness of approximation for H-free edge modification problems. *ACM Transactions on Computation Theory*, 10(2):9:1–9:32, 2018. doi:10.1145/3196834. 10

[31] Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems—a survey. *Discrete Applied Mathematics*, 160(12):1680–1690, 2012. doi:10.1016/j.dam.2012.03.029. 33

[32] Hans L. Bodlaender. On disjoint cycles. *International Journal of Foundations of Computer Science*, 5(1):59–68, 1994. doi:10.1142/S0129054194000049. 11

[33] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219. 8, 278

[34] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009. doi:10.1016/J.JCSS.2009.04.001. 7

[35] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016. doi:10.1137/130947374. 8

[36] Hans L. Bodlaender, John R. Gilbert, Ton Kloks, and Hjálmtyr Hafsteinsson. Approximating treewidth, pathwidth, and minimum elimination tree height. In *Proc. of the 17th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 570 of *LNCS*, pages 1–12, 1991. doi:10.1007/3-540-55121-2_1. 36, 174

[37] Hans L. Bodlaender, Carla Groenland, and Michal Pilipczuk. Parameterized complexity of binary CSP: vertex cover, treedepth, and related parameters. In *Proc. of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 261 of *LIPIcs*, pages 27:1–27:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ICALP.2023.27. 21

[38] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. *SIAM Journal on Discrete Mathematics*, 27(4):2108–2142, 2013. doi:10.1137/120903518. 11

[39] Marthe Bonamy, Konrad K. Dabrowski, Carl Feghali, Matthew Johnson, and Daniël Paulusma. Independent feedback vertex set for p_5-free graphs. *Algorithmica*, 81(4):1342–1369, 2019. doi:10.1007/S00453-018-0474-X. 19, 130

[40] Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Parameterized vertex deletion problems for hereditary graph classes with a block property. In *Proc. of the 42nd International Workshop on Graphs-Theoretic Concepts in Computer Science (WG)*, volume 9941 of *Lecture Notes in Computer Science*, pages 233–244, 2016. doi:10.1007/978-3-662-53536-3_20. 14

[41] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *Journal of the ACM*, 69(1):3:1–3:46, 2022. doi:10.1145/3486655. 24

[42] Henry Roy Brahana. Systems of circuits on two-dimensional manifolds. *Annals of Mathematics*, 23(2):144–168, 1921. doi:10.2307/1968030. 12

[43] Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016. doi:10.1007/s00453-015-0045-3. ix, 17, 22

[44] Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica*, 79(1):139–158, 2017. doi:10.1007/s00453-016-0235-7. ix, 22, 170

[45] Jonathan F. Buss and Judy Goldsmith. Nondeterminism within P. *SIAM Journal on Computing*, 22(3):560–572, 1993. doi:10.1137/0222038. 7

[46] Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Algorithms for the edge-width of an embedded graph. *Computational Geometry*, 45(5-6):215–224, 2012. doi:10.1016/J.COMGEO.2011.12.002. 142

[47] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6. 14

[48] Leizhen Cai and Yufei Cai. Incompressibility of h-free edge modification problems. *Algorithmica*, 71(3):731–757, 2015. doi:10.1007/S00453-014-9937-X. 14

[49] Rutger Campbell, J. Pascal Gollin, Kevin Hendrey, and Sebastian Wiederrecht. Odd-Minors II: Bipartite treewidth, 2023. Manuscript under preparation (private communication). 29, 30, 37, 202

[50] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, 73(1):63–86, 2015. doi:10.1007/S00453-014-9904-6. 11

[51] Paul A. Catlin. Hajós' graph-coloring conjecture: Variations and counterexamples. *Journal of Combinatorial Theory, Series B*, 26(2):268–274, 1979. doi:10.1016/0095-8956(79)90062-5. 18

[52] Kevin Cattell and Michael J. Dinneen. A characterization of graphs with vertex cover up to five. In *Proc. of Orders, Algorithms, and Applications, International Workshop ORDAL*, volume 831 of *LNCS*, pages 86–99. Springer, 1994. doi:10.1007/BFB0019428. 115

[53] Kevin Cattell, Michael J. Dinneen, Rodney G. Downey, Michael R. Fellows, and Michael A. Langston. On computing graph minor obstruction sets. *Theoretical Computer Science*, 233:107–127, 2000. doi:10.1016/S0304-3975(97)00300-9. 46

[54] L Sunil Chandran and Fabrizio Grandoni. Refined memorisation for vertex cover. In *Proc. of the 1st International Workshop of Parameterized and Exact Computation (IWPEC)*, pages 61–70. Springer, 2004. doi:10.1007/978-3-540-28639-4_6. 7

[55] Dimitris Chatzidimitriou, Dimitrios M. Thilikos, and Dimitris Zoros. Sparse obstructions for minor-covering parameters. *Discrete Applied Mathematics*, 278:28–50, 2020. doi:10.1016/j.dam.2019.10.021. 17, 190, 191

[56] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *Journal of the ACM*, 63(5):40:1–40:65, 2016. doi:10.1145/2820609. 26

[57] Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *Journal of Computer and System Sciences*, 74(7):1188–1198, 2008. doi:10.1016/J.JCSS.2008.05.002. 11, 17

[58] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *Journal of Algorithms*, 41(2):280–301, 2001. doi:10.1006/JAGM.2001.1186. 7, 114

[59] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved parameterized upper bounds for vertex cover. In *Proc. of the 31st International Symposium of the Mathematical Foundations of Computer Science (MFCS)*, volume 4162 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2006. doi:10.1007/11821069_21. vi, 7, 17, 115

[60] Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM Journal on Computing*, 45(4):1171–1229, 2016. doi:10.1137/15M1032077. 45, 233, 249

[61] Benny Chor, Mike Fellows, and David W. Juedes. Linear kernels in linear time, or how to save k colors in $o(n^2)$ steps. In *Proc. of the 30th International Workshop on Graph-Theoretic*

Concepts in Computer Science (WG), volume 3353 of *Lecture Notes in Computer Science*, pages 257–269. Springer, 2004. doi:10.1007/978-3-540-30559-0_22. 7

[62] Maria Chudnovsky, Gérard Cornuéjols, Xinming Liu, Paul D. Seymour, and Kristina Vuskovic. Recognizing berge graphs. *Combinatorica*, 25(2):143–186, 2005. doi:10.1007/S00493-005-0012-8. 261

[63] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics (2)*, 164(1):51–229, 2006. doi:10.4007/annals.2006.164.51. 271

[64] Julia Chuzhoy. Improved Bounds for the Flat Wall Theorem. In *Proc. of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 256–275, 2015. doi:10.1137/1.9781611973730.20. 45

[65] Julia Chuzhoy and Zihan Tan. Towards tight(er) bounds for the Excluded Grid Theorem. *Journal of Combinatorial Theory, Series B*, 146:219–265, 2021. doi:10.1016/j.jctb.2020.09.010. 26, 89

[66] Marc Comas and Maria Serna. Vertex fusion under distance constraints. *European Journal of Combinatorics*, 30(7):1612–1623, 2009. doi:10.1016/j.ejc.2009.03.014. 19, 33

[67] Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H. ix, 8, 24, 43, 47, 54, 132, 276

[68] Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2012. URL: http://www.cambridge.org/fr/knowledge/isbn/item5758776/?site_locale=fr_FR. 231, 258

[69] Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Computer Science Review*, 48:100556, 2023. doi:10.1016/j.cosrev.2023.100556. 10, 14, 19, 129

[70] Radu Curticapean. Counting perfect matchings in graphs that exclude a single-crossing minor. *CoRR*, abs/1406.4056, 2014. arXiv:1406.4056. 263

[71] Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In *Proc. of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms(SODA)*, pages 1650–1669. SIAM, 2016. doi:10.1137/1.9781611974331.CH113. 263

[72] Radu Curticapean and Mingji Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod 2k. In *Proc. of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 994–1009. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.65. 263

[73] Marek Cygan. Deterministic parameterized connected vertex cover. In *Proc. of the 13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 7357 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2012. doi:10.1007/978-3-642-31155-0_9. 19

[74] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3. 5, 7, 266

[75] Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. Hitting forbidden subgraphs in graphs of bounded treewidth. *Information and Computation*, 256:62–82, 2017. doi:10.1016/J.IC.2017.04.009. 14

[76] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Transactions on Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707. 11

[77] Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *Proc. of the 22nd IEEE Symposium on Logic in Computer Science (LICS)*, pages 270–279. IEEE Computer Society, 2007. doi:10.1109/LICS.2007.31. 46

[78] Davi de Andrade, Júlio Araújo, Laure Morelle, Ignasi Sau, and Ana Silva. On the parameterized complexity of computing good edge-labelings. *CoRR*, abs/2408.15181, 2024. arXiv:2408.15181, doi:10.48550/ARXIV.2408.15181. 9

[79] Frank K. H. A. Dehne, Michael R. Fellows, Michael A. Langston, Frances A. Rosamond, and Kim Stevens. An $\mathrm{O}(2^{\mathrm{O}(\mathrm{k})}\mathrm{n}^3)$ FPT algorithm for the undirected feedback vertex set problem. *Theory of Computing Systems*, 41(3):479–492, 2007. doi:10.1007/S00224-007-1345-Z. 11

[80] Frank K. H. A. Dehne, Michael R. Fellows, Frances A. Rosamond, and Peter Shaw. Greedy localization, iterative compression, modeled crown reductions: New FPT techniques, an improved algorithm for set splitting, and a novel 2k kernelization for vertex cover. In *Proc. of First International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 3162 of *Lecture Notes in Computer Science*, pages 271–280. Springer, 2004. doi:10.1007/978-3-540-28639-4_24. 7

[81] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005. doi:10.1145/1101821.1101823. 26, 142

[82] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensional parameters and local treewidth. *SIAM Journal on Discrete Mathematics*, 18(3):501–511, 2004. doi:10.1137/S0895480103433410. 72, 107, 238, 267

[83] Erik D Demaine and Mohammad Taghi Hajiaghayi. Bidimensionality: new connections between FPT algorithms and PTASs. In *Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 5, pages 590–601, 2005. doi:10.5555/1070432.1070514. 27

[84] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS))*, pages 637–646. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.14. 27

[85] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Decomposition, approximation, and coloring of odd-minor-free graphs. In *Proc. of the 21st Annual ACM-SIAM*

Symposium on Discrete Algorithms (SODA), pages 329–344. SIAM, 2010. doi:10.1137/1.9781611973075.28. xi, 18, 29, 201

[86] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. The bidimensional theory of bounded-genus graphs. *SIAM Journal on Discrete Mathematics*, 20(2):357–371, 2006. doi:10.1137/040616929. 133, 142

[87] Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, 5th edition, 2017. doi:10.1007/978-3-662-53622-3. 2, 51, 72, 93, 98, 140, 144, 172, 215

[88] Reinhard Diestel, Ken-ichi Kawarabayashi, Theodor Müller, and Paul Wollan. On the excluded minor structure theorem for graphs of large tree-width. *Journal of Combinatorial Theory, Series B*, 102(6):1189–1210, 2012. doi:10.1016/j.jctb.2012.07.001. 72, 93, 94

[89] Öznur Yaşar Diner, Archontia C. Giannopoulou, Giannos Stamoulis, and Dimitrios M. Thilikos. Block elimination distance, 2021. arXiv:2103.01872. 17

[90] Michael J. Dinneen. Too many minor order obstructions (for parameterized lower ideals). *Journal of Universal Computer Science*, 3(11):1199–1206, 1997. doi:10.3217/jucs-003-11. 34

[91] Michael J. Dinneen, Kevin Cattell, and Michael R. Fellows. Forbidden minors to graphs with small feedback sets. *Discrete Mathematics*, 230(1-3):215–252, 2001. doi:10.1016/S0012-365X(00)00083-2. 17

[92] Michael J. Dinneen and Rongwei Lai. Properties of vertex cover obstructions. *Discrete Mathematics*, 307(21):2484–2500, 2007. doi:10.1016/j.disc.2007.01.003. 17, 34, 275

[93] Rodney G. Downey and Michael R. Fellows. Fixed-parameter intractability. In *Proc. of the 7th Annual Structure in Complexity Theory Conference*, pages 36–49. IEEE Computer Society, 1992. doi:10.1109/SCT.1992.215379. 5

[94] Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness III: some structural aspects of the W hierarchy. In *Complexity Theory: Current Research, Dagstuhl Workshop*, pages 191–225. Cambridge University Press, 1992. 5, 11

[95] Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM Journal on Computing*, 24(4):873–921, 1995. doi:10.1137/S0097539792228228. 5, 6

[96] Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theoretical Computer Science*, 141(1&2):109–131, 1995. doi:10.1016/0304-3975(94)00097-3. 5, 6

[97] Rodney G. Downey and Michael R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, 1999. doi:10.1007/978-1-4612-0515-9. 6

[98] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1. 5

[99] Maël Dumas, Anthony Perez, Mathis Rocton, and Ioan Todinca. Polynomial kernels for edge modification problems towards block and strictly chordal graphs. *CoRR*, abs/2201.13140, 2022. arXiv:2201.13140. 129

[100] Zdenek Dvorák, Archontia C. Giannopoulou, and Dimitrios M. Thilikos. Forbidden graphs for tree-depth. *European Journal of Combinatorics*, 33(5):969–979, 2012. doi:10.1016/j.ejc.2011.09.014. 37, 188

[101] Zdenek Dvorák and Robin Thomas. List-coloring apex-minor-free graphs. *CoRR*, 2014. arXiv:1401.1399. xi, 28, 48, 72, 93

[102] Zdenek Dvorák and Paul Wollan. A structure theorem for strong immersions. *Journal of Graph Theory*, 83(2):152–163, 2016. doi:10.1002/JGT.21990. 29

[103] Walther Dyck. Beiträge zur Analysis situs: I. Aufsatz. Ein-und zweidimensionale Mannigfaltigkeiten. *Mathematische Annalen*, 32:457–512, 1888. doi:10.1007/BF01443580. 12

[104] Eduard Eiben, Robert Ganian, Thekla Hamm, and O-joung Kwon. Measuring what matters: A hybrid approach to dynamic programming with treewidth. *Journal of Computer and System Sciences*, 121:57–75, 2021. doi:10.1016/j.jcss.2021.04.005. ix, 22, 201, 265, 267

[105] David Eppstein and Vijay V. Vazirani. NC algorithms for computing a perfect matching and a maximum flow in one-crossing-minor-free graphs. *SIAM Journal on Computing*, 50(3):1014–1033, 2021. doi:10.1137/19M1256221. 263

[106] Joshua Erde. A unified treatment of linked and lean tree-decompositions. *Journal of Combinatorial Theory, Series B*, 130:114–143, 2018. doi:10.1016/j.jctb.2017.12.001. 190

[107] Paul Erdös and George Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935. doi:10.1007/978-0-8176-4842-8_3. 74

[108] Paul Erdős and Lajos Pósa. On independent circuits contained in a graph. *Canadian Journal of Mathematics*, 17:347–352, 1965. doi:10.4153/CJM-1965-035-8. 120, 123

[109] Alastair Farrugia. Vertex-partitioning into fixed additive induced-hereditary properties is NP-hard. *Electronic Journal of Combinatorics*, 11(1), 2004. doi:10.37236/1799. 10, 21, 279

[110] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008. doi:10.1137/05064299X. 8

[111] Michael R. Fellows. Blow-ups, win/win's, and crown rules: Some new directions in FPT. In *Proc. of the 29th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 2880 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2003. doi:10.1007/978-3-540-39890-5_1. 7

[112] Michael R. Fellows, Danny Hermelin, and Frances A. Rosamond. Well quasi orders in subclasses of bounded treewidth graphs and their algorithmic applications. *Algorithmica*, 64(1):3–18, 2012. doi:10.1007/S00453-011-9545-Y. 9

[113] Michael R. Fellows and Michael A. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the ACM*, 35(3):727–739, 1988. doi:10.1145/44483.44491. 17

[114] Michael R. Fellows and Michael A. Langston. An analogue of the myhill-nerode theorem and its use in computing finite-basis characterizations (extended abstract). In *Proc. of the 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 520–525. IEEE Computer Society, 1989. doi:10.1109/SFCS.1989.63528. 17

[115] Michael R. Fellows and Michael A. Langston. On search, decision, and the efficiency of polynomial-time algorithms. *Journal of Computer and System Sciences*, 49(3):769–779, 1994. doi:10.1016/S0022-0000(05)80079-0. 17

[116] Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Frances A. Rosamond, and Saket Saurabh. Graph layout problems parameterized by vertex cover. In *Proc. of the 19th International Symposium on Algorithms and Computation (ISAAC)*, volume 5369 of *Lecture Notes in Computer Science*, pages 294–305. Springer, 2008. doi:10.1007/978-3-540-92182-0_28. 9

[117] Jirí Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science*, 412(23):2513–2523, 2011. doi:10.1016/J.TCS.2010.10.043. 9

[118] Michael E Fisher. Statistical mechanics of dimers on a plane lattice. *Physical Review*, 124(6):1664, 1961. doi:10.1103/PhysRev.124.1664. 262

[119] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X. 5, 6

[120] Fedor V. Fomin, Petr A. Golovach, Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. Compound logics for modification problems. *ACM Transactions on Computational Logic*, 26(1):2:1–2:57, 2025. doi:10.1145/3696451. 24, 26, 35, 38, 46, 68, 234, 275

[121] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Modification to planarity is fixed parameter tractable. In *Proc. of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *LIPIcs*, pages 28:1–28:17, 2019. doi:10.4230/LIPIcs.STACS.2019.28. ix, xii, 19, 20, 35, 46, 125, 276

[122] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. On the parameterized complexity of graph modification to first-order logic properties. *Theory of Computing Systems*, 64(2):251–271, 2020. doi:10.1007/S00224-019-09938-8. 10, 23

[123] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Parameterized complexity of elimination distance to first-order logic properties. *ACM Transactions on Computational Logic*, 23(3):17:1–17:35, 2022. doi:10.1145/3517129. 22

[124] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and kernelization. *SIAM Journal on Discrete Mathematics*, 30(1):383–410, 2016. doi:10.1137/140997889. 17

[125] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar $\mathcal{F}$-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *Proc. of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012. doi:10.1109/FOCS.2012.62. 17

[126] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting topological minors is FPT. In *Proc. of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1317–1326, 2020. doi:10.1145/3357713.3384318. 14, 46, 279

[127] Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Excluded grid minors and efficient polynomial-time approximation schemes. *Journal of the ACM*, 65(2):10:1–10:44, 2018. doi:10.1145/3154833. 267

[128] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, Michal Pilipczuk, and Marcin Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. *ACM Transactions on Algorithms*, 14(3):34:1–34:45, 2018. doi:10.1145/3186898. 8

[129] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M Thilikos. Bidimensionality and kernels. In *Proc. of the 21st annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 503–510. SIAM, 2010. doi:10.1137/1.9781611973075.43. 27

[130] Fedor V. Fomin, Saket Saurabh, and Neeldhara Misra. Graph modification problems: A modern perspective. In *Proc. of the 9th International Workshop on Frontiers in Algorithmics (FAW)*, volume 9130 of *LNCS*, pages 3–6. Springer, 2015. doi:10.1007/978-3-319-19647-3_1. 9

[131] Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Large induced subgraphs via triangulations and CMSO. *SIAM Journal on Computing*, 44(1):54–87, 2015. doi:10.1137/140964801. 8

[132] George K Francis and Jeffrey R Weeks. Conway's zip proof. *The American mathematical monthly*, 106(5):393–399, 1999. doi:10.1080/00029890.1999.12005061. 12

[133] Harvey Friedman, Neil Robertson, and Paul D. Seymour. The metamathematics of the graph minor theorem. In *Logic and combinatorics*, volume 65, pages 229–261. AMS, 1987. doi:10.1090/conm/065/891251. 17

[134] Hanna Furmanczyk, Marek Kubale, and Stanislaw P. Radziszowski. On bipartization of cubic graphs by removal of an independent set. *Discrete Applied Mathematics*, 209:115–121, 2016. doi:10.1016/J.DAM.2015.10.036. 130

[135] Anna Galluccio and Martin Loebl. On the theory of pfaffian orientations. i. perfect matchings and permanents. *The Electronic Journal of Combinatorics*, 6, 1999. doi:10.37236/1438. 263

[136] Cyril Gavoille and Claire Hilaire. Minor-universal graph for graphs on surfaces. *CoRR*, abs/2305.06673, 2023. arXiv:2305.06673, doi:10.48550/ARXIV.2305.06673. 71, 106

[137] Jim Geelen, Bert Gerards, Bruce A. Reed, Paul D. Seymour, and Adrian Vetta. On the odd-minor variant of Hadwiger's conjecture. *Journal of Combinatorial Theory, Series B*, 99(1):20–29, 2009. doi:10.1016/j.jctb.2008.03.006. 18, 279

[138] Archontia C. Giannopoulou, Ken-ichi Kawarabayashi, Stephan Kreutzer, and O-joung Kwon. The Directed Flat Wall Theorem. In *Proc. of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 239–258, 2020. doi:10.1137/1.9781611975994.15. 45

[139] Archontia C. Giannopoulou, Michal Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, and Marcin Wrochna. Cutwidth: Obstructions and algorithmic aspects. *Algorithmica*, 81(2):557–588, 2019. doi:10.1007/s00453-018-0424-7. 18, 191

[140] Archontia C. Giannopoulou, Michal Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, and Marcin Wrochna. Linear kernels for edge deletion problems to immersion-closed graph classes. *SIAM Journal on Discrete Mathematics*, 35(1):105–151, 2021. doi:10.1137/18M1228839. 18

[141] Archontia C. Giannopoulou and Dimitrios M. Thilikos. Optimizing the graph minors weak structure theorem. *SIAM Journal on Discrete Mathematics*, 27(3):1209–1227, 2013. doi:10.1137/110857027. 45, 233

[142] Petr A. Golovach, Marcin Kaminski, Spyridon Maniatis, and Dimitrios M. Thilikos. The parameterized complexity of graph cyclability. *SIAM Journal on Discrete Mathematics*, 31(1):511–541, 2017. doi:10.1137/141000014. 46

[143] Petr A. Golovach, Marcin Kaminski, Daniël Paulusma, and Dimitrios M. Thilikos. Induced packing of odd cycles in planar graphs. *Theoretical Computer Science*, 420:28–35, 2012. doi:10.1016/j.tcs.2011.11.004. 46

[144] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Combing a linkage in an annulus. *SIAM Journal on Discrete Mathematics*, 37(4):2332–2364, 2023. doi:10.1137/22M150914X. 68

[145] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Hitting topological minor models in planar graphs is fixed parameter tractable. *ACM Transactions on Algorithms*, 19(3):23:1–23:29, 2023. doi:10.1145/3583688. 14

[146] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Model-checking for first-order logic with disjoint paths predicates in proper minor-closed graph classes. In *Proc. of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3684–3699. SIAM, 2023. doi:10.1137/1.9781611977554.CH141. 46, 68, 234

[147] Petr A. Golovach, Pim van 't Hof, and Daniël Paulusma. Obtaining planarity by contracting few edges. *Theoretical Computer Science*, 476:38–46, 2013. doi:10.1016/J.TCS.2012.12.041. 19, 129

[148] Maximilian Gorsky, Michał T Seweryn, and Sebastian Wiederrecht. Polynomial bounds for the graph minor structure theorem. *CoRR*, abs/2504.02532, 2025. arXiv:2504.02532, doi:10.48550/arXiv.2504.02532. 27, 274

[149] Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23(4):613–632, 2003. doi:10.1007/S00493-003-0037-9. 267

[150] Martin Grohe. Quasi-4-connected components. In *Proc. of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 55 of *LIPIcs*, pages 8:1–8:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.8. 254, 280

[151] Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *Proc. of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 479–488. ACM, 2011. doi:10.1145/1993636.1993700. 46

[152] Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics - AMS-ASL Joint Special Session*, volume 558 of *Contemporary Mathematics*, pages 181–206. American Mathematical Society, 2009. URL: https://web.archive.org/web/20120901072630id_/http://www.cs.technion.ac.il:80/~janos/RESEARCH/AMS-Book-files/pdfs/05_GroheK.pdf. 24

[153] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095. 24

[154] Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM Journal on Computing*, 44(1):114–159, 2015. doi:10.1137/120892234. 29

[155] Martin Grötschel, László Lovász, and Alexander Schrijver. Polynomial algorithms for perfect graphs. In *North-Holland mathematics studies*, volume 88, pages 325–356. Elsevier, 1984. doi:10.1016/S0304-0208(08)72943-8. 262, 266

[156] Martin Grötschel and William R. Pulleyblank. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters*, 1(1):23–27, 1981. doi:10.1016/0167-6377(81)90020-1. 198, 227

[157] Andrzej Grzesik, Tereza Klimosová, Marcin Pilipczuk, and Michal Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *ACM Transactions on Algorithms*, 18(1):4:1–4:57, 2022. doi:10.1145/3414473. 266

[158] Bertrand Guenin. A characterization of weakly bipartite graphs. *Journal of Combinatorial Theory, Series B*, 83(1):112–168, 2001. doi:10.1006/jctb.2001.2051. 198, 227

[159] Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386–1396, 2006. doi:10.1016/j.jcss.2006.02.001. 11

[160] Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *Proc. of the 1st International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 3162 of *LNCS*, pages 162–173, 2004. doi:10.1007/978-3-540-28639-4_15. vii, 9, 17

[161] Arvind Gupta and Russell Impagliazzo. Bounding the size of planar intertwines. *SIAM Journal on Discrete Mathematics*, 10(3):337–358, 1997. doi:10.1137/S0895480192239931. 17

[162] Gregory Z. Gutin, Mark Jones, and Magnus Wahlström. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM Journal on Discrete Mathematics*, 30(4):2177–2205, 2016. doi:10.1137/15M1034337. 21

[163] Hugo Hadwiger. Über eine Klassifikation der Streckenkomplexe. *Vierteljschr. Naturforsch. Ges. Zürich*, 88(2):133–142, 1943. URL: https://www.ngzh.ch/archiv/1943_88/88_2/88_17.pdf. 18

[164] Pinar Heggernes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. *Theoretical Computer Science*, 511:172–180, 2013. doi:10.1016/J.TCS.2012.03.013. 10, 270, 271

[165] Pinar Heggernes, Pim van 't Hof, Benjamin Lévêque, Daniel Lokshtanov, and Christophe Paul. Contracting graphs to paths and trees. *Algorithmica*, 68(1):109–132, 2014. doi:10.1007/S00453-012-9670-2. 34, 35, 129, 275, 276

[166] Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. *SIAM Journal on Discrete Mathematics*, 27(4):2143–2156, 2013. doi:10.1137/130907392. 46

[167] Micha Hofri. *Analysis of algorithms: Computational methods and mathematical tools*. Oxford University Press, Inc., 1995. 4

[168] Huynh, Tony. *The Linkage Problem for Group-labelled Graphs*. PhD thesis, University of Waterloo, 2009. URL: http://hdl.handle.net/10012/4716. viii, 18

[169] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774. 276

[170] Yoichi Iwata. Linear-time kernelization for feedback vertex set. In *Proc. of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80 of *LIPIcs*, pages 68:1–68:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.ICALP.2017.68. 11

[171] Lars Jaffke, Laure Morelle, Ignasi Sau, and Dimitrios M. Thilikos. Dynamic programming on bipartite tree decompositions. In *Proc. of the 18th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 285 of *LIPIcs*, pages 26:1–26:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.IPEC.2023.26. 38, 210, 226, 278

[172] Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory of Computing Systems*, 53(2):263–299, 2013. doi:10.1007/S00224-012-9393-4. 11

[173] Bart M. P. Jansen and Jari J. H. de Kroon. FPT algorithms to compute the elimination distance to bipartite graphs and more. In *Proc. of the 47th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 12911 of *LNCS*, pages 80–93, 2021. doi:10.1007/978-3-030-86838-3_6. 22

[174] Bart M. P. Jansen, Jari J. H. de Kroon, and Michał Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proc. of the 53rd Annual ACM-SIGACT Symposium on Theory of Computing (STOC)*, pages 1757–1769, 2021. doi:10.1145/3406325.3451068. 14, 21, 46, 265, 267

[175] Bart M. P. Jansen and Stefan Kratsch. On polynomial kernels for structural parameterizations of odd cycle transversal. In *Parameterized and Exact Computation - 6th International Symposium, IPEC 2011, Saarbrücken, Germany, September 6-8, 2011. Revised Selected Papers*, volume 7112 of *Lecture Notes in Computer Science*, pages 132–144. Springer, 2011. doi:10.1007/978-3-642-28050-4_11. 11

[176] Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014. doi:10.1137/1.9781611973402.130. 11, 17

[177] Tommy R Jensen and Bjarne Toft. *Graph coloring problems*. Wiley, 2011. doi:10.1002/9781118032497. 18

[178] Marcin Kaminski and Naomi Nishimura. Finding an induced path of given parity in planar graphs in polynomial time. In *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 656–670, 2012. doi:10.1137/1.9781611973099.55. 46

[179] Marcin Kamiński and Dimitrios M. Thilikos. Contraction checking in graphs on surfaces. In *Proc. of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 14 of *LIPIcs*, pages 182–193, 2012. doi:10.4230/LIPIcs.STACS.2012.182. 46

[180] Iyad A. Kanj, Michael J. Pelsmajer, and Marcus Schaefer. Parameterized algorithms for feedback vertex set. In *Proc. of the 1st International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 3162 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2004. doi:10.1007/978-3-540-28639-4_21. 11

[181] Richard M. Karp. Reducibility among combinatorial problems. In *Proc. of a symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9. vi, 4, 6, 114, 227

[182] Pieter W Kasteleyn. The statistics of dimers on a lattice: I. the number of dimer arrangements on a quadratic lattice. *Physica*, 27(12):1209–1225, 1961. doi:10.1016/0031-8914(61)90063-5. 262

[183] Pieter W. Kasteleyn. Graph theory and crystal physics. *Graph theory and theoretical physics*, pages 43–110, 1967. 264

[184] Ken-ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–648, 2009. doi:10.1109/FOCS.2009.45. 17

[185] Ken-ichi Kawarabayashi. Totally odd subdivisions and parity subdivisions: Structures and coloring. In *Proc. of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1013–1029. SIAM, 2013. doi:10.1137/1.9781611973105.73. 29

[186] Ken-ichi Kawarabayashi and Yusuke Kobayashi. The edge disjoint paths problem in Eulerian graphs and 4-edge-connected graphs. In *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 345–353, 2010. doi:10.1137/1.9781611973075.29. 46

[187] Ken-ichi Kawarabayashi and Yusuke Kobayashi. Linear min-max relation between the treewidth of an H-minor-free graph and its largest grid minor. *Journal of Combinatorial Theory, Series B*, 141:165–180, 2020. doi:10.1016/j.jctb.2019.07.007. 61

[188] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce A. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012. doi:10.1016/j.jctb.2011.07.004. 16, 45, 56, 121

[189] Ken-ichi Kawarabayashi, Stephan Kreutzer, and Bojan Mohar. Linkless and flat embeddings in 3-space. *Discrete & Computational Geometry*, 47(4):731–755, 2012. doi:10.1007/s00454-012-9413-9. 46

[190] Ken-ichi Kawarabayashi, Zhentao Li, and Bruce A. Reed. Recognizing a Totally Odd K_4-subdivision, Parity 2-disjoint Rooted Paths and a Parity Cycle Through Specified Elements. In *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 318–328, 2010. doi:10.1137/1.9781611973075.27. 46

[191] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width.

In *Proc. of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 771–780, 2008. doi:10.1109/FOCS.2008.53. 46

[192] Ken-ichi Kawarabayashi and Bruce A. Reed. An (almost) linear time algorithm for odd cyles transversal. In *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 365–378. SIAM, 2010. doi:10.1137/1.9781611973075.31. 11, 18, 29

[193] Ken-ichi Kawarabayashi, Bruce A. Reed, and Paul Wollan. The graph minor algorithm with parity conditions. In *Proc. of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 27–36. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.52. viii, 18, 279

[194] Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. A new proof of the flat wall theorem. *Journal of Combinatorial Theory, Series B*, 129:204–238, 2018. doi:10.1016/j.jctb.2017.09.006. 27, 43, 45, 48, 58, 63, 89, 131, 233, 238

[195] Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. Quickly excluding a non-planar graph. *CoRR*, abs/2010.12397, 2021. arXiv:2010.12397. 27, 46, 48, 72, 73, 75, 78, 79, 80, 81, 86, 87, 93, 94, 274

[196] Ken-ichi Kawarabayashi and Paul Wollan. A Shorter Proof of the Graph Minor Algorithm: The Unique Linkage Theorem. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 687–694, 2010. doi:10.1145/1806689.1806784. 68

[197] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016. doi:10.1145/2797140. 17

[198] Valerie King, S. Rao, and Robert Endre Tarjan. A faster deterministic maximum flow algorithm. *Journal of Algorithms*, 17(3):447–474, 1994. doi:10.1006/jagm.1994.1044. 196, 215, 219, 225

[199] Martin Knor. Characterization of minor-closed pseudosurfaces. *Ars Combinatoria*, 43, 1996. URL: https://www.math.sk/knor/PREP/p009.pdf. 32, 273

[200] Yasuaki Kobayashi and Hisao Tamaki. Treedepth parameterized by vertex cover number. In *Proc. of the 11th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 63 of *LIPIcs*, pages 18:1–18:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.IPEC.2016.18. 9

[201] Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Information Processing Letters*, 114(10):556–560, 2014. doi:10.1016/j.ipl.2014.05.001. 11

[202] Tomasz Kociumaka and Marcin Pilipczuk. Deleting Vertices to Graphs of Bounded Genus. *Algorithmica*, 81(9):3655–3691, 2019. doi:10.1007/s00453-019-00592-7. 13, 17, 276

[203] Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *Proc. of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–192, 2021. doi:10.1109/FOCS52979.2021.00026. 8, 54

[204] Tuukka Korhonen and Daniel Lokshtanov. An improved parameterized algorithm for treewidth. In *Proc. of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 528–541. ACM, 2023. doi:10.1145/3564246.3585245. 8, 278

[205] Tuukka Korhonen, Michal Pilipczuk, and Giannos Stamoulis. Minor containment and disjoint paths in almost-linear time. In *Proc. of the 65th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 53–61, 2024. doi:10.1109/FOCS61266.2024.00014. vii, 16, 22, 35, 121, 122, 156, 233, 238, 276

[206] Jan Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52(3):233–252, 1994. doi:10.1016/0166-218X(94)90143-0. 268

[207] Stefan Kratsch and Pascal Schweitzer. Isomorphism for graphs of bounded feedback vertex set number. In *Proc. of the 12th Scandinavian Symposium and Workshops on Algorithm TheoryAlgorithm Theory (SWAT)*, volume 6139 of *Lecture Notes in Computer Science*, pages 81–92. Springer, 2010. doi:10.1007/978-3-642-13731-0_9. 11

[208] Stephan Kreutzer. Algorithmic meta-theorems. In *Proc. of the 3rd International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 10–12. Springer, 2008. 24

[209] R. Krithika, V. K. Kutty Malu, Roohani Sharma, and Prafullkumar Tale. Parameterized complexity of biclique contraction and balanced biclique contraction. In *Proc. of the 43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 284 of *LIPIcs*, pages 8:1–8:18, 2023. doi:10.4230/LIPICS.FSTTCS.2023.8. 275

[210] R. Krithika, Pranabendu Misra, and Prafullkumar Tale. A single exponential-time FPT algorithm for cactus contraction. *Theoretical Computer Science*, 954:113803, 2023. doi:10.1016/J.TCS.2023.113803. 35, 129

[211] Casimir Kuratowski. Sur le probleme des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930. 15

[212] Jens Lagergren. Efficient parallel algorithms for graphs of bounded tree-width. *Journal of Algorithms*, 20(1):20–44, 1996. doi:10.1006/JAGM.1996.0002. 8

[213] Jens Lagergren. Upper bounds on the size of obstructions and intertwines. *Journal of Combinatorial Theory, Series B*, 73:7–40, 1998. doi:10.1006/jctb.1997.1788. 17, 191

[214] Michael Lampis. A kernel of order 2 k-c log k for vertex cover. *Information Processing Letters*, 111(23-24):1089–1091, 2011. doi:10.1016/J.IPL.2011.09.003. 7

[215] Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/S00453-011-9554-X. 9

[216] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4. 10, 228

[217] Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^\star(2.7k)$ time. *ACM Transactions on Algorithms*, 18(4):34:1–34:26, 2022. doi:10.1145/3504027. 11

[218] Shaohua Li and Marcin Pilipczuk. An improved FPT algorithm for independent feedback vertex set. *Theory of Computing Systems*, 64(8):1317–1330, 2020. doi:10.1007/S00224-020-09973-W. 19, 35, 130, 276

[219] Wenjun Li, Qilong Feng, Jianer Chen, and Shuai Hu. Improved kernel results for some FPT problems based on simple observations. *Theoretical Computer Science*, 657:20–27, 2017. doi:10.1016/J.TCS.2016.06.012. 35, 129, 276

[220] Wenjun Li and Binhai Zhu. A $2k$-kernelization algorithm for vertex cover based on crown decomposition. *Theoretical Computer Science*, 739:80–85, 2018. doi:10.1016/J.TCS.2018.05.004. 7

[221] Carlos V. G. C. Lima, Dieter Rautenbach, Uéverton S. Souza, and Jayme Luiz Szwarcfiter. On the computational complexity of the bipartizing matching problem. *Annals of Operations Research*, 316(2):1235–1256, 2022. doi:10.1007/S10479-021-03966-9. 19, 130

[222] Carlos Vinícius G. C. Lima, Dieter Rautenbach, Uéverton S. Souza, and Jayme Luiz Szwarcfiter. Decycling with a matching. *Information Processing Letters*, 124:26–29, 2017. doi:10.1016/J.IPL.2017.04.003. 130

[223] Daniel Lokshtanov. Wheel-Free Deletion Is W[2]-Hard. In *Proc. of the 3rd International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 5018 of *LNCS*, pages 141–147, 2008. doi:10.1007/978-3-540-79723-4_14. 10, 271

[224] Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Reducing CMSO model checking to highly connected graphs. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming, (ICALP)*, volume 107 of *LIPIcs*, pages 135:1–135:14, 2018. doi:10.4230/LIPICS.ICALP.2018.135. 45, 231, 279, 280

[225] Daniel Lokshtanov, Saket Saurabh, and Somnath Sikdar. Simpler parameterized algorithm for OCT. In *Proc. of 20th International Workshop on Combinatorial Algorithms (IWOCA)*, volume 5874 of *Lecture Notes in Computer Science*, pages 380–384. Springer, 2009. doi:10.1007/978-3-642-10217-2_37. 11

[226] Konrad Majewski, Michał Pilipczuk, and Marek Sokołowski. Maintaining cmso2 properties on dynamic structures with bounded feedback vertex number. *ACM Transactions on Computation Theory*, 17(2):1–72, 2025. doi:10.1145/3715884. 11

[227] Rafael Martí, Panos M. Pardalos, and Mauricio G. C. Resende. *Handbook of Heuristics*. Springer, 2018. doi:10.1007/978-3-319-07124-4. 4

[228] Dániel Marx and R. B. Sandeep. Incompressibility of H-free edge modification problems: Towards a dichotomy. *Journal of Computer and System Sciences*, 125:25–58, 2022. doi:10.1016/J.JCSS.2021.11.001. 14

[229] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012. doi:10.1007/s00453-010-9484-z. 17, 46, 47, 132

[230] Thomas W. Mattman and Mike Pierce. The K_{n+5} and $K_{3^2,1^n}$ families are obstructions to n-apex. In *Knots, links, spatial graphs, and algebraic invariants*, volume 689 of *Contemporary Mathematics*, pages 137–158. American Mathematical Society, 2017. doi:10.1090/conm/689. 17

[231] Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{|V|}|E|)$ Algorithm for Finding Maximum Matching in General Graphs. In *Proc. of the 21st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 17–27. IEEE Computer Society, 1980. doi:10.1109/SFCS.1980.12. 215

[232] Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal on Discrete Mathematics*, 12(1):6–26, 1999. doi:10.1137/S089548019529248X. 144

[233] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins series in the mathematical sciences. Johns Hopkins University Press, 2001. URL: http://jhupbooks.press.jhu.edu/ecom/MasterServlet/GetItemDetailsHandler?iN=9780801866890&qty=1&source=2&viewMode=3&loggedIN=false&JavaScript=y. 11, 17, 106, 133

[234] Hendrik Molter, Meirav Zehavi, and Amit Zivan. Treewidth parameterized by feedback vertex number. *CoRR*, abs/2504.18302, 2025. arXiv:2504.18302. 11

[235] Laure Morelle, Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. Faster parameterized algorithms for modification problems to minor-closed classes. *TheoretiCS*, 3, 2024. doi:10.46298/THEORETICS.24.19. 35, 128, 145, 276

[236] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995. doi:10.1017/CBO9780511814075. 4

[237] N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. LP can be a cure for parameterized problems. In *Proc. of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 14 of *LIPIcs*, pages 338–349. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPICS.STACS.2012.338. 7

[238] George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(1):232–248, 1975. doi:10.1007/BF01580444. 7

[239] Rian Neogi, M. S. Ramanujan, Saket Saurabh, and Roohani Sharma. On the parameterized complexity of deletion to \(\boldsymbol{\mathcal{H}}\)-free strong components. *SIAM Journal on Discrete Mathematics*, 38(4):3079–3110, 2024. doi:10.1137/23M1548591. 14

[240] Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4. 21

[241] Jaroslav Nesetril and Patrice Ossona de Mendez. On low tree-depth decompositions. *Graphs Comb.*, 31(6):1941–1963, 2015. doi:10.1007/S00373-015-1569-7. 21

[242] Rolf Niedermeier and Peter Rossmanith. Upper bounds for vertex cover further improved. In *Proc. of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1563 of *Lecture Notes in Computer Science*, pages 561–570. Springer, 1999. doi:10.1007/3-540-49116-3_53. 7

[243] Rolf Niedermeier and Peter Rossmanith. On efficient fixed-parameter algorithms for weighted vertex cover. *J. Algorithms*, 47(2):63–77, 2003. doi:10.1016/S0196-6774(03)00005-1. 7

[244] Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. Parameterized counting algorithms for general graph covering problems. In *Proc. of the 9th International Workshop on Algorithms and Data Structures (WADS)*, pages 99–109, 2005. doi:10.1007/11534273_10. 17

[245] Yoshio Okamoto, Ryuhei Uehara, and Takeaki Uno. Counting the number of matchings in chordal and chordal bipartite graph classes. In *Graph-Theoretic Concepts in Computer Science, 35th International Workshop, WG 2009, Montpellier, France, June 24-26, 2009. Revised Papers*, volume 5911 of *Lecture Notes in Computer Science*, pages 296–307, 2009. doi:10.1007/978-3-642-11409-0_26. 263

[246] James B. Orlin. Max flows in $O(nm)$ time, or better. In *Proc. of the 45th annual ACM Symposium on Theory of Computing Conference (STOC)*, pages 765–774. ACM, 2013. doi:10.1145/2488608.2488705. 196, 215, 219, 225

[247] Christophe Paul, Evangelos Protopapas, and Dimitrios M Thilikos. Graph Parameters, Universal Obstructions, and WQO. *CoRR*, abs/2304.03688, 2023. URL: https://arxiv.org/abs/2304.03688, arXiv:2304.03688. 120

[248] Christophe Paul, Evangelos Protopapas, and Dimitrios M. Thilikos. Universal obstructions of graph parameters, 2023. arXiv:2304.14121. 120

[249] Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa. *DNA Computing - New Computing Paradigms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1998. doi:10.1007/978-3-662-03563-4. 4

[250] Ljubomir Perkovic and Bruce A. Reed. An improved algorithm for finding tree decompositions of small width. *International Journal of Foundations of Computer Science*, 11(3):365–371, 2000. doi:10.1142/S0129054100000247. 165

[251] Fábio Protti and Uéverton S. Souza. Decycling a graph by the removal of a matching: new algorithmic and structural aspects in some classes of graphs. *Discrete Mathematics & Theoretical Computer Science*, 20(2), 2018. doi:10.23638/DMTCS-20-2-15. 19, 130

[252] Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for undirected feedback vertex set. In *Proc. of the 13th International Symposium on Algorithms and Computation (ISAAC)*, volume 2518 of *Lecture Notes in Computer Science*, pages 241–248. Springer, 2002. doi:10.1007/3-540-36136-7_22. 11

[253] Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for finding feedback vertex sets. *ACM Transactions on Algorithms*, 2(3):403–415, 2006. doi:10.1145/1159892.1159898. 11

[254] Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009. 11, 18, 198, 224

[255] Bruce A. Reed. Finding approximate separators and computing tree width quickly. In *Proc. of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 221–228. ACM, 1992. doi:10.1145/129712.129734. 8

[256] Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth. In *Proc. of the 41st International Colloquium on*

Automata, Languages, and Programming (ICALP), volume 8572 of *LNCS*, pages 931–942, 2014. doi:10.1007/978-3-662-43948-7_77. 21, 36, 47, 159, 174, 175, 176

[257] Neil Robertson, Daniel P. Sanders, Paul D. Seymour, and Robin Thomas. Efficiently four-coloring planar graphs. In *Proc. of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 571–575. ACM, 1996. doi:10.1145/237814.238005. 262

[258] Neil Robertson and Paul D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5. 16

[259] Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3. 16, 265, 266

[260] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4. 8, 16

[261] Neil Robertson and Paul D. Seymour. Graph Minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(2):92–114, 1986. doi:10.1016/0095-8956(86)90030-4. xi, 16, 25, 26, 71, 89, 190, 191

[262] Neil Robertson and Paul D. Seymour. Graph minors. VI. Disjoint paths across a disc. *Journal of Combinatorial Theory, Series B*, 41(1):115–138, 1986. doi:10.1016/0095-8956(86)90031-6. 16

[263] Neil Robertson and Paul D. Seymour. Graph minors. VII. Disjoint paths on a surface. *Journal of Combinatorial Theory, Series B*, 45(2):212–254, 1988. doi:10.1016/0095-8956(88)90070-6. 16

[264] Neil Robertson and Paul D. Seymour. Graph minors. IV. Tree-width and well-quasi-ordering. *Journal of Combinatorial Theory, Series B*, 48(2):227–254, 1990. doi:10.1016/0095-8956(90)90120-O. 16

[265] Neil Robertson and Paul D. Seymour. Graph minors. IX. Disjoint crossed paths. *Journal of Combinatorial Theory, Series B*, 49(1):40–77, 1990. doi:10.1016/0095-8956(90)90063-6. 16, 58, 94

[266] Neil Robertson and Paul D. Seymour. Graph minors. VIII. A kuratowski theorem for general surfaces. *Journal of Combinatorial Theory, Series B*, 48(2):255–288, 1990. doi:10.1016/0095-8956(90)90121-F. 16

[267] Neil Robertson and Paul D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991. doi:10.1016/0095-8956(91)90061-N. 16

[268] Neil Robertson and Paul D. Seymour. Excluding a graph with one crossing. In *Graph structure theory*, volume 147 of *Contemporary Mathematics*, pages 669–675. American Mathematical Society, 1993. doi:10.1090/conm/147. 28, 40, 71, 91, 93

[269] Neil Robertson and Paul D. Seymour. Graph Minors. XI. Circuits on a surface. *Journal of Combinatorial Theory, Series B*, 60(1):72–106, 1994. doi:10.1006/JCTB.1994.1007. 16, 142

[270] Neil Robertson and Paul D. Seymour. Graph minors. XII. Distance on a surface. *Journal of Combinatorial Theory, Series B*, 64(2):240–272, 1995. doi:10.1006/JCTB.1995.1034. 16

[271] Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006. vii, x, xiii, 8, 16, 24, 38, 43, 45, 46, 63, 88, 89, 121, 131, 132, 145, 233, 234, 238, 276

[272] Neil Robertson and Paul D. Seymour. Graph minors. XIV. Extending an embedding. *Journal of Combinatorial Theory, Series B*, 65(1):23–50, 1995. doi:10.1006/JCTB.1995.1042. 16

[273] Neil Robertson and Paul D. Seymour. Graph minors. XV. Giant steps. *Journal of Combinatorial Theory, Series B*, 68(1):112–148, 1996. doi:10.1006/JCTB.1996.0059. 16

[274] Neil Robertson and Paul D. Seymour. Graph minors. XVII. Taming a vortex. *Journal of Combinatorial Theory, Series B*, 77(1):162–210, 1999. doi:10.1006/JCTB.1999.1919. 16

[275] Neil Robertson and Paul D. Seymour. Graph Minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X. x, xi, 16, 27, 48

[276] Neil Robertson and Paul D. Seymour. Graph minors. XVIII. Tree-decompositions and well-quasi-ordering. *Journal of Combinatorial Theory, Series B*, 89(1):77–108, 2003. doi:10.1016/S0095-8956(03)00067-4. 16

[277] Neil Robertson and Paul D. Seymour. Graph minors. XIX. Well-quasi-ordering on a surface. *Journal of Combinatorial Theory, Series B*, 90(2):325–385, 2004. doi:10.1016/J.JCTB.2003.08.005. 16

[278] Neil Robertson and Paul D. Seymour. Graph Minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. doi:10.1016/j.jctb.2004.08.001. vii, 16, 22, 35, 53, 122

[279] Neil Robertson and Paul D. Seymour. Graph Minors. XXI. Graphs with unique linkages. *Journal of Combinatorial Theory, Series B*, 99(3):583–616, 2009. doi:10.1016/j.jctb.2008.08.003. 16, 68, 121

[280] Neil Robertson and Paul D. Seymour. Graph minors. XXIII. Nash-williams' immersion conjecture. *Journal of Combinatorial Theory, Series B*, 100(2):181–205, 2010. doi:10.1016/J.JCTB.2009.07.003. 16

[281] Neil Robertson and Paul D. Seymour. Graph Minors. XXII. Irrelevant vertices in linkage problems. *Journal of Combinatorial Theory, Series B*, 102(2):530–563, 2012. doi:10.1016/j.jctb.2007.12.007. 16, 68, 121

[282] Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, 1994. doi:10.1006/jctb.1994.1073. 71, 142, 150

[283] Ignasi Sau and Uéverton dos Santos Souza. Hitting forbidden induced subgraphs on bounded treewidth graphs. *Information and Computation*, 281:104812, 2021. doi:10.1016/J.IC.2021.104812. 14

[284] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. II. parameterized algorithms. *ACM Transactions on Algorithms*, 18(3):21:1–21:30, 2022. doi:10.1145/3519028. vii, xii, 17, 35, 46, 68, 127, 131, 132, 145, 165

[285] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. I. Bounding the obstructions. *Journal of Combinatorial Theory, Series B*, 161:180–227, 2023. doi:10.1016/J.JCTB.2023.02.012. vii, 37, 46, 48, 66, 67, 68, 132, 145, 146, 147, 150, 159, 163, 191

[286] Ignasi Sau, Giannos Stamoulis, and Dimitrios M Thilikos. A more accurate view of the flat wall theorem. *Journal of Graph Theory*, 107(2):263–297, 2024. doi:10.1002/jgt.23121. 43, 45, 46, 60, 62, 64, 65, 68, 131, 145, 163, 233, 238, 243

[287] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. Parameterizing the quantification of CMSO: model checking on minor-closed graph classes. In *Proc. of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3728–3742, 2025. doi:10.1137/1.9781611978322.124. x, xii, 24, 25, 26, 35, 38, 46, 127, 128, 234, 275, 276

[288] Alexander Schrijver. Disjoint circuits of prescribed homotopies in a graph on a compact surface. *Journal of Combinatorial Theory, Series B*, 51(1):127–159, 1991. doi:10.1016/0095-8956(91)90009-9. 279

[289] Alexander Schrijver. Disjoint homotopic paths and trees in a planar graph. *Discrete and Computational Geometry*, 6:527–574, 1991. doi:10.1007/BF02574704. 279

[290] Paul D. Seymour. A bound on the excluded minors for a surface, March 1994. Manuscript. URL: https://web.math.princeton.edu/~pds/papers/surfacebound/bound.pdf. 17

[291] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004. doi:10.1016/J.DAM.2004.01.007. 10

[292] Roded Sharan. *Graph Modification Problems and their Applications to Genomic Research*. PhD thesis, Sackler Faculty of Exact Sciences, School of Computer Science, 2002. URL: http://www.cs.tau.ac.il/thesis/thesis/Roded-Sharan-phd.pdf. 9

[293] Jozef Sirán and Pavol Gvozdjak. Kuratowski-type theorems do not extend to pseudosurfaces. *Journal of Combinatorial Theory, Series B*, 54(2):209–212, 1992. doi:10.1016/0095-8956(92)90052-Y. 32

[294] Arezou Soleimanfallah and Anders Yeo. A kernel of order 2k-c for vertex cover. *Discrete Mathematics*, 311(10-11):892–895, 2011. doi:10.1016/J.DISC.2011.02.014. 7

[295] Stamoulis, Giannos. *Logics and Algorithms for Graph Minors*. PhD thesis, University of Montpellier, 2023. URL: https://www.irif.fr/~stamoulis/2023_STAMOULIS_thesis_final.pdf. 24

[296] Ulrike Stege and Michael Ralph Fellows. An improved fixed parameter tractable algorithm for vertex cover. *Technical report/Departement Informatik, ETH Zürich*, 318, 1999. URL: https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/69332/eth-4359-01.pdf. 7

[297] Raphael Steiner. Improved bound for improper colourings of graphs with no odd clique minor. *Combinatorics, Probability and Computing*, 32(2):326–333, 2023. doi:10.1017/S0963548322000268. 18

[298] Simon Straub, Thomas Thierauf, and Fabian Wagner. Counting the number of perfect matchings in K 5-free graphs. *Theory of Computing Systems*, 59(3):416–439, 2016. doi:10.1007/S00224-015-9645-1. 263

[299] Siamak Tazari. Faster approximation schemes and parameterized algorithms on (odd-)h-minor-free graphs. *Theoretical Computer Science*, 417:95–107, 2012. doi:10.1016/j.tcs.2011.09.014. xi, 29, 201

[300] Harold NV Temperley and Michael E Fisher. Dimer problem in statistical mechanics-an exact result. *Philosophical Magazine*, 6(68):1061–1063, 1961. doi:10.1080/14786436108243366. 262

[301] Dimitrios M. Thilikos. Graph minors and parameterized algorithm design. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *LNCS*, pages 228–256. Springer, 2012. doi:10.1007/978-3-642-30891-8_13. 46

[302] Dimitrios M. Thilikos and Sebastian Wiederrecht. Excluding surfaces as minors in graphs, 2023. arXiv:2306.01724v5. x, 12, 25, 29, 31, 32, 49, 72, 94, 105, 106, 107, 108, 110, 273

[303] Dimitrios M. Thilikos and Sebastian Wiederrecht. Killing a vortex. *Journal of the ACM*, 71(4):27:1–27:56, 2024. doi:10.1145/3664648. 72, 93, 263, 265, 274

[304] Robin Thomas. A Menger-like property of tree-width: The finite case. *Journal of Combinatorial Theory, Series B*, 48(1):67–76, 1990. doi:10.1016/0095-8956(90)90130-R. 190, 191

[305] Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001. doi:10.1006/jctb.2000.2013. 53

[306] Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2):32:1–32:8, 2010. doi:10.1145/1721837.1721848. 11

[307] Dekel Tsur. Faster deterministic algorithm for co-path set. *Information Processing Letters*, 180:106335, 2023. doi:10.1016/J.IPL.2022.106335. 35, 129, 276

[308] Leslie G. Valiant. Holographic algorithms. *SIAM Journal on Computing*, 37(5):1565–1594, 2008. doi:10.1137/070682575. 263

[309] Klaus Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114:570–590, 1937. doi:10.1007/BF01594196. 15

[310] Hassler Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55(1):245–254, 1933. doi:10.2307/2371127. 144

[311] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. URL: http://www.cambridge.org/de/knowledge/isbn/item5759340/?site_locale=de_DE. 4

[312] Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018. doi:10.1016/J.JCSS.2017.11.003. 21

[313] Mihalis Yannakakis. Node- and Edge-Deletion NP-Complete Problems. In *Proc. of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, 1978. doi:10.1145/800133.804355. 10

[314] Mihalis Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10(2):310–327, 1981. doi:10.1137/0210022. 38, 195, 228, 278

[315] Noson S. Yanofsky. An introduction to quantum computing. In *Proof, Computation and Agency - Logic at the Crossroads*, volume 352 of *Synthese library*, pages 145–180. Springer, 2011. doi:10.1007/978-94-007-0080-2_10. 4

[316] Dimitris Zoros. *Obstructions and Algorithms for Graph Layout Problems*. PhD thesis, National and Kapodistrian University of Athens, Department of Mathematics, July 2017. URL: http://users.uoa.gr/~dzoros/Files/thesis.pdf. 17