THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE

DE MATHÉMATIQUES ET D'INFORMATIQUE

SPÉCIALITÉ : INFORMATIQUE

Par **Timothé PICAVET**

Algorithmes distribués et structures locales dans les graphes

Sous la direction de : **Cyril Gavoille** (directeur)
et **Marthe Bonamy** (directrice)

Soutenue le 23 juin 2026

Membres du jury :

Mme. Marthe BONAMY	Directrice de recherche	Université de Bordeaux, CNRS	Directrice
M. Nicolas BOUSQUET	Chargé de recherche	Université Lyon 1, CNRS	Rapporteur
Mme. Maria CHUDNOVSKY	Professor	Princeton University	Examinatrice
M. Cyril GAVOILLE	Professeur	Université de Bordeaux	Directeur
M. Christoph LENZEN	Professor	Rekyjavik University	Rapporteur
M. Jean-François MARCKERT	Directeur de Recherche	Université de Bordeaux	Président
M. Nati LINIAL	Professor	The Hebrew University of Jerusalem	Examinateur
Mme. Alexandra WESOLEK	Chargée de recherche	Université de Bordeaux	Invitée

Algorithmes distribués et structures locales dans les graphes

Résumé : Dans cette thèse, nous nous intéressons aux algorithmes distribués du modèle LOCAL qui fonctionnent en temps constant, c'est-à-dire dont le nombre de rondes est indépendant du nombre de sommets, appliquées à l'approximation de problèmes de domination au sein de différentes classes de graphes. Nous donnons des algorithmes d'approximation de MDS sur des familles de graphes paramétrées, dont le facteur d'approximation ne dépend pas du paramètre. Dans un premier temps, nous prouvons qu'étant donné un algorithme LOCAL produisant une bonne approximation sur les graphes planaires, peut être transformé en un algorithme LOCAL produisant une bonne approximation sur les graphes plongeables dans une surface de genre eulérien. En nous appuyant sur l'algorithme de Heydt et al., nous en déduisons une approximation de facteur $34 + \varepsilon$ pour les graphes de genre borné. Ce résultat améliore considérablement l'état de l'art précédent de $24g + O(1)$ établi par Amiri et al., ainsi que le facteur de $91 + \varepsilon$ obtenu par Czygrinow et al. dans le cas particulier des surfaces orientables. Nous généralisons ensuite ce résultat dans deux directions : d'une part, en considérant d'autres problèmes de graphes étudiés en algorithmique distribuée, et d'autre part en étendant nos résultats à des classes de graphes au-delà du genre borné. Nous prouvons ces résultats via une série de méta-théorèmes portant sur certains problèmes de minimisation. Dans un second temps, nous montrons que certains graphes structurés (qui excluent un mineur H de pathwidth au plus 2) admettent un algorithme distribué déterministe en $f(H)$ rondes calculant une 50-approximation pour le problème MDS. Bien que des algorithmes distribués rapides et approchés pour ces problèmes fussent déjà connus pour les graphes sans mineur H, tous présentaient un facteur d'approximation dépendant de H. Un nouvel ingrédient clé dans l'analyse de ces différents algorithmes distribués est l'utilisation de la dimension asymptotique, une notion géométrique introduite par Gromov en 1993.
Mots-clés : Théorie des graphes, Algorithmes distribués, Structures interdites

---

Distributed algorithms and local structures in graphs

Abstract: In this thesis, we study constant-time distributed algorithms in the LOCAL model, that is, algorithms whose number of rounds is independent of the number of vertices, applied to the approximation of domination problems within various graph classes. We provide approximation algorithms for MDS on parameterized graph families whose approximation factor does not depend on the parameter. First, we prove that a LOCAL algorithm that produces a good approximation on planar graphs can be transformed into a LOCAL algorithm that produces a good approximation on graphs embeddable in a surface of Euler genus. Building on the algorithm of Heydt et al., we derive a $(34 + \varepsilon)$-approximation for bounded-genus graphs. This result significantly improves the previous state of the art, namely the $24g + O(1)$ bound established by Amiri et al., as well as the $91 + \varepsilon$ factor obtained by Czygrinow et al. in the particular case of orientable surfaces. We then generalize this result in two directions: first, by considering other graph problems studied in distributed algorithms, and second, by extending our results to graph classes beyond bounded genus. We prove these results through a series of meta-theorems concerning certain minimization problems. Second, we show that certain structured graphs (namely those excluding a minor H of pathwidth at most 2), admit a deterministic distributed algorithm running in $f(H)$ rounds that computes a 50-approximation for the MDS problem. Although fast distributed approximation algorithms for these problems were already known for H-minor-free graphs, all of them had an approximation factor depending on H. A new key ingredient in the analysis of these various distributed algorithms is the use of asymptotic dimension, a geometric notion introduced by Gromov in 1993.
Keywords: Graph theory, Distributed algorithms, Forbidden structures

---

Unité de recherche

LaBRI, Université de Bordeaux (UMR 5800) 351, cours de la Libération F-33405 Talence cedex

Contents

Acknowledgements

As I'm writing this, I've defended my thesis and I'm about to leave Bordeaux. It feels strange, but in a good way! I certainly couldn't have finished this PhD without all the people who supported me and shared these years with me. A PhD life isn't always easy, but you made this one a time in my life that I truly enjoyed. I hope I forget no one here![1]

[1] If I did, please consider it an indexing error, not a peer-reviewed assessment of our friendship. Anyone I've forgotten will appear in the journal version.

Tout d'abord, je dois énormément à Marthe et Cyril, qui m'ont encadré pendant ces trois super années de thèse. Vos conseils, en recherche comme en dehors, m'ont été précieux. Si cette thèse s'est aussi bien passée, c'est en grande partie grâce à vous ! Votre aide et votre présence dans les moments plus difficiles ont compté, par exemple lorsque vous m'avez encouragé à parler du syndrome de l'imposteur.

I want to thank everybody who agreed to be in my jury: Nicolas Bousquet, Maria Chudnovsky, Christoph Lenzen, Jean-François Marckert, Nati Linial, and Alexandra Wesolek. I really couldn't have asked for a better jury.

I'm also very grateful to Alexandra W, Alexandre B, Arnaud, Édouard, Jean-François, Martín, Théo, Tomas, and Pedro for all those wonderful research visits. Some visits produced papers, and the others produced excellent reasons to organize another visit. Édouard, je n'oublierai pas que tu m'as recueilli à Lyon alors que j'y passais. Théo, je suis très heureux que tu m'aies accompagné dans mes recherches et jusqu'au Chili. Je suis sûr qu'un jour, nous réussirons à utiliser la compression d'entropie ![2]

[2] Oui, c'est une menace.

To my coauthors[3] Alex S, Alkida, Avinandan, Binh-Minh, Cyril, Daniel, Dennis, Édouard, Étienne, Fabian, Gustav, Henrik, Jukka, Julien B, Laure, Lucas, Mamadou, Marthe, Martín, Mathieu M, Michał, Ngoc-Trung, Nicolas T, Pedro, Pierre, Samuel, Sebastian, Théo, Théotime, Tim and Ugo: I had a blast doing research with all of you.

[3] Please do not forget to breathe while reading this list.

I'm especially grateful to Michał, Mathieu M, Jukka, and Alexandra for mentoring me before and during the early days of my PhD.

Merci aussi à Mathieu H, qui m'a mentoré au LaBRI. Et l'autre Mathieu H (totalement pas du tout la même personne) mérite une mention spéciale pour avoir organisé ces superbes sessions de JdR.

Xavier, j'ai beaucoup aimé enseigner à l'IOGS avec toi ces deux dernières années, c'était une chouette expérience. Je souhaite aussi remercier tout le personnel du LaBRI, et en particulier Anthony G, Claire D, Emmanuelle L, Isabelle G et Ophélie L pour leur aide administrative précieuse.

Il y a eu beaucoup de gens très sympas au LaBRI et en dehors. Vous avez tous largement contribué à rendre ces années aussi chouettes ! À suivre, une liste non exhaustive, mais déjà relue plus de fois que certains chapitres.

Je pense tout d'abord à Edgar, Émile, Rémi et Taïssir (le Papa-pingouin-chelbel et la Scooby-team avec Quentin évidemment). Je pourrais fournir davantage de contexte, mais ça ne rendrait probablement pas ce nom plus raisonnable. Il y aurait tant de choses à raconter ! Tous ces $\Omega(1)$ événements et ces moments passés ensemble comptent parmi mes meilleurs souvenirs, tellement nombreux qu'on ne pourrait pas les compter sur les doigts des mains (même en ajoutant ceux des pieds, et ceux de vous tous réunis. Après ça, même les mathématiques recommandent d'arrêter de compter). J'ai eu beaucoup de chance de vous connaître à Bordeaux. On a fait une super équipe, et vous avez eu un énorme impact sur moi. Je ne l'oublierai jamais !

Edgar, j'ai eu énormément de chance de t'avoir comme cobureau. Tu es vraiment une super personne, et j'ai beaucoup aimé passer tous ces moments avec toi, même si tu m'as annihilé sur Smash Ultimate. J'avais pourtant appuyé très fort sur tous les boutons. Tu m'as aussi beaucoup aidé à grandir et montré que rien n'était impossible, pas même manger trois pizzas en un seul repas. J'espère que tu arriveras à finir ton jeu un jour. Tu ne peux désormais plus abandonner : cet objectif apparaît dans un document académique officiel.

Émile, merci pour tous ces moments passés ensemble à faire de la musique (ou, selon certains témoins, du bruit), à faire des blagues ou à regarder des films à l'envers (une méthode pas très efficace pour éviter les spoilers, ni pour comprendre l'intrigue. Heureusement, il n'y en avait pas dans Koyaanisqatsi). Je n'oublierai pas ton talent pour l'impro. Peu de gens peuvent se vanter d'avoir donné autant de profondeur psychologique à une simple vague.

Rémi, je garde de super souvenirs de tout ce qu'on a partagé : les glaces à deux boules[4], les concerts à l'Auditorium et toutes ces soirées. J'espère qu'un jour tu réussiras à jouer du violon sans agacer tes voisins.[5]

Taïssir, ta joie de vivre, les pauses thé, tous les événements partagés (les game jams, les miams et tout le reste) ainsi que nos conversations, par exemple sur la façon de réussir à aimer l'art qu'on produit, ont énormément compté. J'ai beaucoup appris, notamment à différencier Massy, Palaiseau, Orsay et Saclay. Quatre endroits que je soupçonnais jusque-là d'être différentes sorties d'une même gare.

En parlant de game jams, vous pouvez d'ailleurs admirer le résultat de nos décisions parfaitement raisonnables : Reverse Heist[6], Assemblée nationale Simulator et Lost in Resoluti0n.

Gabriel, je n'oublierai ni les miams ni le fait que tu m'aies accompagné au Mexique pour ce PODC mémorable en plein cyclone[7]. Tout le monde devrait une fois dans sa vie surveiller la trajectoire d'un cyclone entre deux exposés.

[4] Deux boules en théorie, mais environ trois tonnes dès qu'il fallait reprendre le vélo. La physique n'a toujours pas fourni d'explication satisfaisante.

[5] Ou alors, il suffit de les convaincre que c'est du néo-jazz microtonal expérimental et qu'ils n'ont simplement pas encore compris ton génie.

[6] Qui raconte un cas incontestable de haute trahison textile. Je vous laisse y jouer pour plus de contexte.

[7] Nous avons ainsi testé la robustesse de la conférence face à un adversaire météorologique

Merci à Timothée (avec œufs[8]) pour ton humour timoth-époustouflant, et à Thibault, pour tous les moments passés à l'intérieur comme en dehors de l'AFoDIB.

Et puisqu'on parle de l'AFoDIB : c'est vraiment une super association. Un grand merci[9] à toutes celleux qui l'ont fait vivre. Impossible de ne pas citer Aline, Amaury, Anna-Louise, Antonio, Arthur, Athénaïs, Aymeric, Bernardo, Clara, Clément L, Colleen, Coralie, Corto, Françoise, Gabriel, Joseph, Joséphine, Juliette, Nacim, Noémie, Oscar, Pierre B, Théo et Yanis, ainsi que toutes les autres personnes déjà citées, avec qui j'ai partagé tant de bons moments au labo comme en dehors.

Je garde de très bons souvenirs des raclettes et des soirées jeux avec Sarah, des soirées lanternes avec Méline et des soirées films avec Romain D, toujours pleines de bonne humeur et bien miamées.

Merci aussi aux membres et aux organisateurs de la GRSF (La Groupade de Recherche en Sport Fondamental, dite la gréssif) ; ils ont eux aussi joyeusement animé cette dernière année avec ces sports endiablés[10].

Merci à Vincent pour les parties d'échecs !

Julien D et Maxime F, c'est toujours fun de vous retrouver aux conférences et de vous parler, de rien tout autant que de sujets personnels qui nous tiennent à cœur.

Fanny, I loved those long walks filled with conversation, the cheese-eating sessions, and all those entertaining (and sometimes slightly absurd) takes. J'espère que tu liras cette phrase en français très facilement.

Ton passage parmi nous, Benjamin, aura au moins permis de soigner ta carence en fromage[11]. Même s'il a été bref, il ne sera pas oublié ! Et je te dois toujours un cours de piano[12].

Romain B (el λomain), j'ai beaucoup apprécié toutes nos conversations intéressantes, sur la recherche, mais aussi celles où on balance des blagues toutes les 30 secondes (une densité asymptotiquement optimale). Romain B et Clément R, je n'oublierai pas les parties de Ricochet Robots, le meilleur jeu de tous les temps[13].

Un grand merci à Jonathan : j'ai passé de très bons moments avec toi, au labo et en dehors. Que tu puisses toujours apporter de la bonne humeur là où tu vas !

Je tiens aussi à remercier le reste de mes adelphes académiques : Claire, Clément et Paul, que ce soit pour leur soutien, pour les moments passés ensemble, ou pour leurs talents de karaoké.

Merci à David d'avoir été mon (super) prof de piano pendant ma dernière année de thèse. J'aime beaucoup ta méthode d'apprentissage, et j'ai vraiment l'impression d'avoir beaucoup progressé[14].

J'ai aussi la chance de pouvoir compter sur mes amis en dehors du labo, dont, entre autres, Amadeus, Bastien, Corentin, Dorian, Eeva, Elias, Francesco, Guilhem, Laurent, Nicolas, Sabrina et Sameep.

Laurent, merci d'avoir marqué mes retours à Paris, et pas que, que ce soit avec de superbes explorations, des moments marrants en ligne ou encore en me sauvant des griffes de la Deutsche Bahn en venant me chercher en voiture jusqu'en Allemagne. Une preuve expérimentale que

[8] Aucune coquille n'a été blessée pendant la rédaction de ce jeu de mots.

[9] L'ordre alphabétique a été choisi parce que l'amitié n'admet pas d'ordre total.

[10] Les courbatures restent à ce jour le seul résultat parfaitement reproductible de ces rencontres.

[11] La carence avait été diagnostiquée sans aucune compétence médicale, parce que je n'étais alors pas docteur. Après plusieurs essais, le traitement au fromage n'a révélé aucun effet secondaire, à part l'envie de poursuivre le traitement.

[12] Cette dette est officiellement reconnue, mais aucune date de remboursement n'est annoncée.

[13] Cette affirmation n'a pas été relue par les pairs, car elle est manifestement vraie.

[14] Mes voisins ont peut-être une estimation différente de mes progrès.

le pouvoir de l'amitié est plus fiable que le réseau ferroviaire allemand.

J'ai beaucoup aimé les soirées quiz avec Pierre V, la soirée jeux avec Maximino et Noémie et les sorties à vélo avec Sylvain. C'est à Quentin que je dois la découverte de Day-O et de ses innombrables[15] remixes plus ou moins intéressants (non, ce n'est pas une attaque personnelle).

Gustav, you were a very nice roommate.

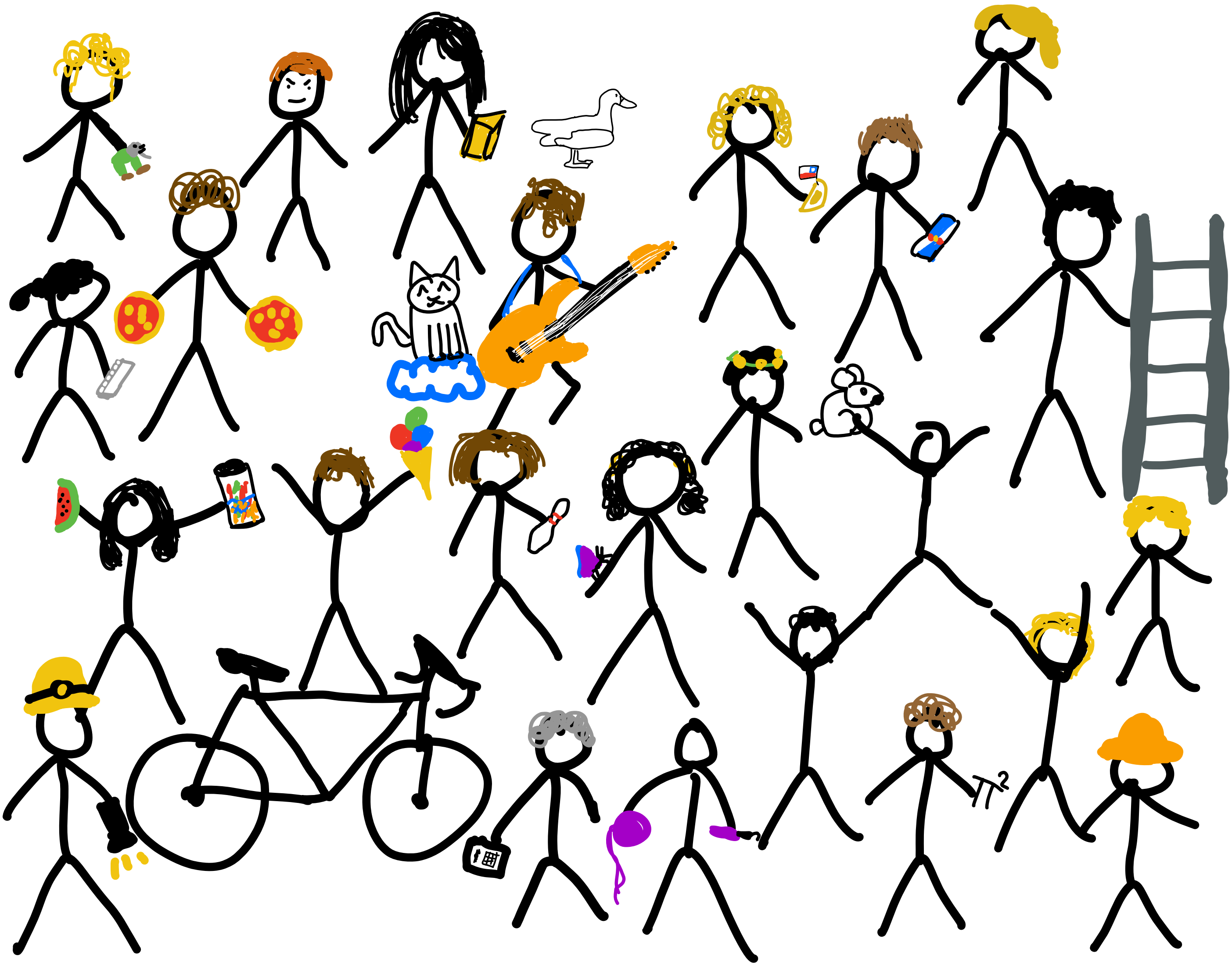
Un énorme merci à Bastien, Dorian (el Dodo), Elias et Elliot, avec qui j'ai partagé de longues conversations, des fous rires et des parties en ligne qui se sont prolongées jusque trèèèès tard[16] dans la nuit. Bastien et Elias, vos visites m'ont fait très plaisir. J'espère pouvoir refaire des randonnées ou du bloc avec vous un jour. Dodo, merci pour les parties de bowling, les crêpes, et les temps records sur les escape games. Merci à Eeva de m'avoir toujours bien accueilli, que ce soit à Londres ou à Paris !

Last but not least, je dois énormément à ma famille, maman, papa, Emily et Elliot, qui ont pris soin de moi et partagé tous ces moments avec moi. Nébula (el gato Nébubu), merci pour tes câlins et tous ces superbes moments. Je ne t'oublierai jamais et je te dédicace cette thèse. Tu es probablement la seule à être passée sur chaque chapitre, principalement en dormant dessus. Merci à Taintaine pour tous ces moments passés ensemble depuis tellement longtemps.

Pas merci à un certain site de synonymes qui m'a suggéré dédommagement, dépendance, grâce, ou encore impitoyable comme synonymes de merci. Je ne lui adresse donc pas toute mon impitoyable.

[15] Le nombre exact de remixes reste inconnu, mais il est conjecturé infini. Et pour cause : après suffisamment de remixes, toute chanson semble converger vers Day-O.

[16] Un grand sage a dit que parfois, le lendemain commence avant le coucher.



Résumé étendu en français

Un *graphe* est constitué de sommets et d'arêtes, chaque arête reliant deux sommets. Cet objet mathématique est très utile pour représenter des réseaux de communication, mais aussi des réseaux routiers, sociaux ou biologiques. Par exemple, dans un réseau de communication d'ordinateurs, on peut représenter chaque ordinateur par un sommet et ses connexions aux autres ordinateurs par des arêtes. Dans un réseau typique, deux ordinateurs ne sont pas toujours reliés directement, et peuvent même être à très grande distance l'un de l'autre. Pour cette raison, tout réseau se heurte à une grande difficulté : centraliser toutes les données en un seul point peut être très lent, voire impossible. Chaque entité doit alors décider à partir d'informations locales : c'est ce qu'on appelle un algorithme distribué. Cette thèse cherche à déterminer quelles propriétés structurelles du réseau rendent possibles des algorithmes distribués à la fois rapides et de bonne qualité.

La difficulté d'un problème distribué ne dépend pas seulement de la taille du réseau, mais aussi de sa structure. Nous en étudions différentes manifestations. Par exemple, le graphe peut être dessinable sur un plan, peut se découper en régions simples, ou alors exclure certaines configurations. Toutes ces propriétés permettent de réduire le nombre de rondes de communication et augmenter la qualité des solutions.

Voir peu, décider vite

Nous travaillons principalement dans le modèle LOCAL : les sommets sont des ordinateurs dotés d'identifiants uniques et les arêtes des liens de communication. Le calcul se déroule en rondes synchrones, avec des messages de taille arbitraire et un calcul local non borné. L'information ne se déplace toutefois que d'une arête par ronde. Donc, après r rondes, la décision d'un sommet ne peut dépendre que de son *voisinage de rayon* r, c'est-à-dire l'ensemble des sommets situés à distance au plus r. Dans un algorithme en un nombre constant de rondes, ce rayon ne grandit pas avec la taille du réseau.

Une grande partie de la thèse porte sur le problème de l'*ensemble dominant minimum*, abrégé MDS pour *Minimum Dominating Set*. Il s'agit de choisir un ensemble D de sommets aussi petit que possible tel que tout sommet appartienne à D ou possède un voisin dans D. Cette formulation modélise, par exemple, le placement de relais dans un réseau. Calculer localement un optimum exact est généralement hors de portée, ou alors très coûteux. Nous cherchons donc des approximations à facteur constant. Une α-approximation renvoie toujours une solution valide, mais sa taille est à un facteur au plus α près de la taille de la solution minimale.

Sur les graphes généraux, aucun algorithme en temps constant ne peut garantir un facteur constant pour MDS. Il faut donc étudier des réseaux dont la structure est plus restreinte. Pour avoir plus de structure, nous interdisons certains *mineurs*. Supprimer des sommets ou des arêtes produit un sous-graphe. Si l'on autorise en plus la contraction d'une arête, c'est-à-dire la fusion de ses deux extrémités, on obtient ce qu'on appelle un mineur. Un graphe est *sans mineur H* lorsqu'aucune suite de ces opérations ne produit le graphe H. Cette restriction décrit de nombreuses familles de graphes structurellement intéressantes.

Des algorithmes planaires aux classes plus générales

Un graphe est *planaire* s'il peut être dessiné dans le plan sans croisement d'arêtes. Certains graphes non planaires peuvent néanmoins être dessinés sans croisement sur une surface plus complexe, par exemple un tore. Le *genre d'Euler g* mesure la complexité minimale d'une surface qui autorise un tel dessin. Cette mesure permet de généraliser les graphes planaires, qui correspondent exactement au cas $g = 0$. Pour le problème de domination, notre premier résultat transforme tout algorithme LOCAL en temps constant et de facteur α sur les graphes planaires en un algorithme de facteur $3\alpha + 1$ sur les graphes de genre d'Euler au plus g. Le nombre de rondes peut dépendre de g, mais pas du nombre n de sommets. En utilisant la meilleure approximation planaire connue, de facteur $11 + \varepsilon$ pour toute constante $\varepsilon > 0$ [HKO+25], nous obtenons ainsi un facteur $34 + \varepsilon$ pour tout genre fixé. Celui-ci est indépendant de g et améliore les résultats précédents.

Pour prouver ce transfert, nous utilisons la notion de *dimension asymptotique*, introduite par Gromov [Gro93], et qui formalise la possibilité de découper certains graphes de manière controlée. Fixons une distance r. Une classe de graphes a dimension asymptotique au plus d si les sommets de chacun de ses graphes peuvent être répartis en $d + 1$ couleurs de sorte qu'une région formée par une même couleur soit de petit diamètre (faible), mais que deux régions différentes de même couleur soient assez lointaines.

Intuitivement, chaque région peut ainsi être entièrement explorée en un nombre de rondes qui ne dépend pas de la taille du réseau. Les couleurs permettent ensuite de réunir les solutions locales tout en limitant le nombre de fois où une même partie de l'optimum intervient dans leur coût. Comme les classes excluant un mineur fixé (incluant les classes de genre borné) ont une dimension asymptotique au plus deux, nous en déduisons les résultats de transfert. Ces résultats ne s'appliquent pas seulement à MDS dans les graphes de genre borné. Nous les généralisons à de plus grandes classes de graphes, et à une classe infinie de problèmes, composée des problèmes “découpables”.

Une seconde contribution concerne les graphes qui excluent $K_{2,t}$ comme mineur, pour un entier $t \geqslant 2$. Ce graphe comporte deux groupes de respectivement 2 et t sommets, avec toutes les arêtes possibles entre les deux groupes et aucune à l'intérieur d'un groupe. Nous donnons un algorithme en $f(t)$ rondes qui calcule une 50-approximation de MDS.

L'algorithme exploite les endroits où le graphe peut être localement séparé : toutes les séparations locales de taille un, et certaines de taille deux, sont sélectionnées dans l'ensemble dominant. Nous montrons ensuite que chaque composante encore non dominée possède un diamètre faible borné. Chaque sommet peut alors en connaître toute la structure et y calculer une solution optimale. Encore une fois, la dimension asymptotique nous permet de borner le nombre total de sommets sélectionnés par rapport à l'optimum global.

La largeur de chemin comme paramètre caché

Ces résultats suggèrent que la taille du mineur interdit n'est pas le bon paramètre pour mesurer l'approximabilité locale de MDS. Le paramètre pertinent est plutôt sa *largeur de chemin*. Pour la comprendre, on peut imaginer parcourir le graphe linéairement en ne gardant en mémoire qu'un petit ensemble de sommets à chaque étape. Ces ensembles, appelés *sacs*, forment une décomposition en chemin : ils couvrent tous les sommets, les deux extrémités de chaque arête apparaissent ensemble dans au moins un sac, et tous les sacs contenant un sommet donné se suivent sans interruption. La largeur de la décomposition est la taille du plus grand sac, moins un. La largeur de chemin, notée $\mathrm{pw}(H)$ pour un graphe H, est la plus petite largeur obtenue parmi toutes ses décompositions. Plus elle est petite, plus le graphe ressemble à un chemin.

Nous prouvons que, pour tout graphe H, aucun algorithme déterministe en $f(H)$ rondes ne peut garantir, sur les graphes sans mineur H, un facteur strictement inférieur à $2\,\mathrm{pw}(H) - 1$ pour MDS. Cette borne motive notre conjecture principale : il devrait exister, sur les graphes sans mineur H, une approximation dont le facteur est au plus une constante universelle fois $\mathrm{pw}(H)$ et dont le nombre de rondes ne dépend que de H. La dépendance en la largeur de chemin serait alors optimale à un facteur constant près.

Nous démontrons le premier cas non trivial de la conjecture : pour tout graphe H de largeur de chemin au plus deux, les graphes sans mineur H admettent une 50-approximation déterministe en un nombre de rondes constant. Pour se faire, nous utilisons le même algorithme que pour le cas des graphes sans mineur $K_{2,t}$, mais avec une analyse plus poussée. Si un certain ensemble d'un graphe a un diamètre faible trop grand, des arguments de connectivité permettraient de construire n'importe quel H de largeur de chemin deux comme mineur. Une fois les séparations locales sélectionnées, le diamètre faible de chaque composante restante est donc borné, ce qui rend possible leur résolution locale exacte.

Le passage d'une largeur de chemin deux à une largeur arbitraire fait apparaître deux difficultés principales. Il faut d'abord distinguer, au sein de coupures locales plus grandes, les sommets qu'il est réellement utile de sélectionner, puis borner leur nombre par rapport à l'optimum. Il faut ensuite montrer que les composantes restantes sont simples à gérer. Cela demande de contrôler à la fois leur connectivité et leur complexité topologique afin d'y forcer l'apparition du mineur interdit.

References

[Gro93] Mikhael Gromov. *Geometric Group Theory: Asymptotic invariants of infinite groups*. Cambridge University Press, 1993.

[HKO+25] Ozan Heydt et al. "Distributed domination on sparse graph classes". In: *European Journal of Combinatorics* 123 (Jan. 2025), p. 103773. DOI: 10.1016/j.ejc.2023.103773.

Part A

Introduction

Chapter I
Background

I.1 A quick introduction to graphs

In this section, we give some basic graph-theoretic definitions. If you are already familiar with basic definitions around graphs, planar graphs, and graph minors, feel free to skip this section.

In this thesis, unless explicitly stated, we consider finite, simple graphs. Let $G = (V, E)$ be a graph where $V(G) := V$ represents the set of *vertices*, and $E(G) := E \subseteq V^2$ represents the set of edges. Two distinct vertices u and v are said to be *adjacent* (or *neighbors*) if the edge $e = uv$ is in $E(G)$. In this case, we also say that the edge e is *incident* to its endpoints u and v.

Some basic graphs. Let us give us some simple examples of graphs. A *path* P_k is a sequence of k distinct vertices where every two consecutive vertices are adjacent (see Figure I.1). The first and last vertex are called the *endpoints*, or *endvertices* of the path, while all its other vertices are its *internal vertices*. The *length* of P corresponds to its number of edges. A path connecting two vertices x and y (in possibly a larger graph) is called an *xy-path*. We extend this notation for sets of vertices. If $A, B \subseteq V(G)$, then a path connecting two vertices $a \in A$ and $b \in B$ is called an AB-path.

Figure I.1: A path graph P_5 with endpoints u and v.

If we instead take a path on $k \geqslant 3$ vertices and add an edge closing the gap between its endpoints, we obtain a *cycle* C_k (see Figure I.2). Cycles are the simplest structures that introduce multiple internally disjoint (disjoint except for their endpoints) paths vertices.

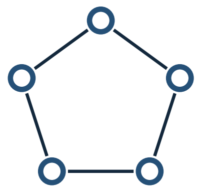


Figure I.2: A cycle graph C_5.

Paths represent the bare minimum of connectivity: they are the most sparse connected structures possible. If we broaden this notion of minimal connectivity to allow branching while strictly prohibiting any cycles, we obtain a *tree*. Formally, a tree is a connected, acyclic graph (see Figure I.3). Because they contain no cycles, trees possess exactly one unique path between any pair of vertices, making them structurally minimal: removing any single edge from a tree immediately disconnects it.

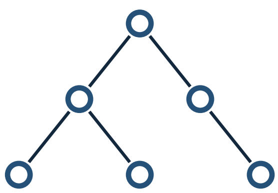


Figure I.3: A tree.

When the graph is a tree, the vertices are often called *nodes*.

At the opposite end of the density spectrum are *cliques*. A clique is a subset of vertices in G such that every two distinct vertices are connected by an edge. The induced subgraph of a clique on n vertices is the *complete graph* K_n (see Figure I.4).

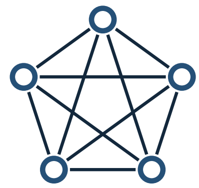


Figure I.4: A complete graph or clique K_5.

These subgraphs often show up, and are linked to the global properties of the supergraph. For instance, Ramsey's Theorem famously says that any sufficiently large graph must contain either a large clique or a large *independent set* (a pairwise non-adjacent set of vertices, i.e. the complement of a clique).

Separators and k-connectivity. A graph G is *connected* if every pair of vertices is joined by a path. More generally, a *connected component* of G is a maximally connected subgraph of G, so that every graph can be written as a disjoint union of its connected components.

As seen before, trees are "structurally fragile", in the sense that removing a single internal vertex may destroy connectivity. This is not the case for well-connected graphs such as cliques. To quantify this, we study *separators*. A subset of vertices $S \subseteq V(G)$ is called a *k-separator* or *k-cut*[1] if the graph $G - S$ has more connected components than G. In particular, when G is connected, this means that $G - S$ is disconnected[2]. Intuitively, separators represent structural bottlenecks in the graph. For every three subsets $X, A, B \subseteq V(G)$, we say that X *separates* A from B in G if no component of $G - X$ contains both a vertex from A and B.

[1] In this thesis, we *do not* study edge separators.

[2] Here we use the notation that $G - S$ is the graph on vertices $V(G) \setminus S$ obtained from G by removing edges incident to vertices not in S.

This leads directly to the concept of *k-connectivity*. A graph G is said to be *k-connected* if it remains connected after the removal of any set of strictly fewer than k vertices. Equivalently, a k-connected graph has no separator of size less than k. Under this framework, trees are only 1-connected, cycles are 2-connected, and the complete graph K_n is $(n-1)$-connected.

Some more structured graphs. A graph G is *bipartite* if its vertex set can be partitioned into two disjoint sets, A and B (its *bi-partition*), such that every edge is incident a vertex in A and a vertex in B. In other words, no two vertices within the same partition are adjacent. This makes bipartite graphs exactly equivalent to *2-colorable* graphs: one can assign one color to all vertices in A and a second color to all vertices in B, guaranteeing that no adjacent vertices share the same color. A classic foundational result states that a graph is bipartite if and only if it contains absolutely no cycles of odd length. When every vertex in A is adjacent to every vertex in B, we obtain a *complete bipartite graph*, denoted $K_{n,m}$ where $n = |A|$ and $m = |B|$.

A distinct but equally interesting graph class is the class of *planar graphs*. A graph is *planar* if it can be drawn on the plane such that no edges intersect, except at their shared vertex endpoints. Such a crossing-free drawing is called a *planar embedding*. Of course, planar graphs naturally appear in the study of road networks, map coloring and the study of printed circuit boards. Interestingly, bipartiteness and planarity intersect in the study of forbidden structures. The two simplest graphs that cannot possibly be drawn on a plane without edge crossings are the dense clique K_5 and the complete bipartite graph $K_{3,3}$ (often famously introduced as the unsolvable "three utilities puzzle", see Section I.1).

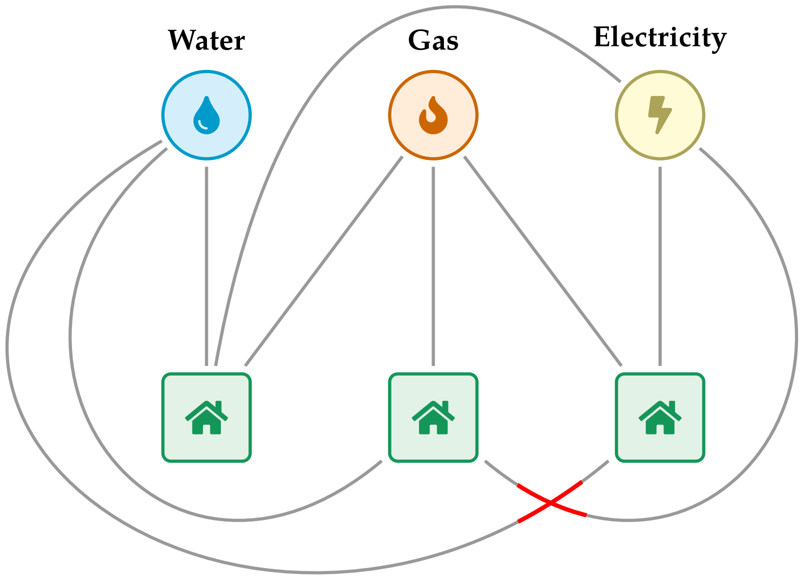


Figure I.5: The Three Utilities Puzzle ($K_{3,3}$). It is impossible to connect all three houses (bottom partition) to all three utilities (top partition) on a 2D plane without at least one edge crossing, making $K_{3,3}$ non-planar.

Subgraphs and unions. A graph H is a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Note that for $u, v \in V(H)$, the edge uv is not necessarily in $E(H)$. Given a subset $S \subseteq V(G)$, the *induced subgraph* $G[S]$ is the graph with vertex set S and edge set containing precisely all edges from $E(G)$ that have both endpoints in S.

Given two disjoint sets X and Y, we denote their union by $X \sqcup Y$ to emphasize their disjointness. When combining two disjoint graphs G_1 and G_2, we denote their *disjoint union* by $G_1 \sqcup G_2$. The resulting graph has vertex set $V(G_1) \sqcup V(G_2)$ and edge set $E(G_1) \sqcup E(G_2)$. For instance, the graph $K_3 \sqcup K_3$ consists of two separate, non-adjacent triangles.

Distances, diameter, and girth. The *distance* $d_G(u, v)$ between two vertices $u, v \in V(G)$ is defined as the number of edges in the shortest path connecting them in G. If no such path exists (meaning u and v are in different connected components), we set $d_G(u, v) = \infty$. We extend this notion to measure the distance between a single vertex v and a subset of vertices $D \subseteq V(G)$ as the length of the shortest path from v to any vertex in D:

$$d_G(v, D) = \min\{d_G(v, u) \mid u \in D\}.$$

Consider a graph G, a vertex v of G, $r \in \mathbb{N}$, and a subset $S \subseteq V(G)$. We denote by $N^r[v]$ the radius-r ball centered at v in G, that is, the set of all vertices at distance at most r from v in G. For short, we denote the *open neighborhood* of v by $N[v] = N^1[v]$, and the *open neighborhood* of v by $N(v) = N[v] \setminus \{v\}$. By extension, $N^r[S] = \bigcup_{v \in S} N^r[v]$. The size of $N(v)$ is called the *degree* of the vertex, and is denoted by $\deg(v)$.

The *diameter* of a graph G, denoted $\operatorname{diam}(G)$, is the maximum distance between any pair of vertices in G. Often, we are interested in how spread out a set S of vertices is within G, rather than within $G[S]$. To capture this, we define the *weak diameter* of a set $S \subseteq V(G)$ as the largest distance in G between two vertices $u, v \in S$. Notice that the weak diameter of S is at most $\operatorname{diam}(G[S])$. It can also be strictly shorter, if the shortest path in G is uses vertices outside of S. Another parameter is the *girth* of a graph, which is the length of its shortest cycle. For example, a tree has no cycles and thus has infinite girth, whereas the famous Petersen graph (see Figure I.6) is well-known for having a girth of 5 and a diameter of 2.

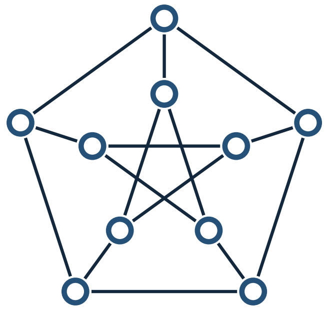


Figure I.6: The Petersen graph.

Graph Minors and Where to Find Them. Graph minors are a cornerstone of structural graph theory. They provide a way to describe when one graph contains the rough macroscopic structure of another. There are two equivalent definitions.

The first one is through edge contractions. A graph H is a *minor* of G if H can be obtained from G by a sequence of three basic operations: deleting a vertex, deleting an edge, and contracting an edge. An edge contraction, denoted G/e with $e = uv \in E(G)$, removes the edge e and merges the vertices u and v into a single new vertex w that is adjacent to all former neighbors of both u and v (see Section I.1).

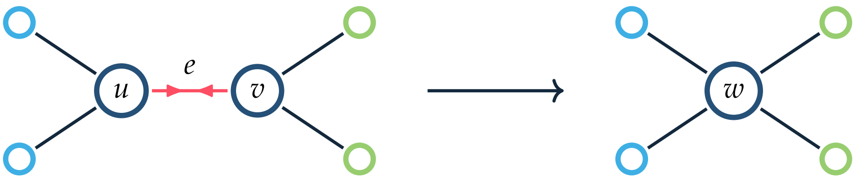


Figure I.7: An illustration of an edge contraction. The red edge $e = uv$ is contracted, merging the vertices u and v into a single new vertex w that is adjacent to all neighbors of both u (in blue) and v (in green).

The second definition is structural and often easier to work with when defining minors. A graph H is a minor of G if there exists a collection of non-empty, pairwise disjoint subsets $\{V_x \mid x \in V(H)\}$ of $V(G)$ such that:

- For every $x \in V(H)$, the induced subgraph $G[V_x]$ is connected.
- For every edge $xy \in E(H)$, there is at least one edge in G with one endpoint in V_x and the other in V_y.

The sets V_x are referred to as the *branch sets* of the minor. Together, they form a *model* of H in G. As an elementary example, any cycle of length $k \geqslant 3$ contains a triangle K_3 as a minor. This is easily seen by contracting $k-3$ edges until only three vertices remain. See Section I.1 for another example.

A class of graphs $\mathcal{C}$ is said to be *minor-closed* if, for every $G \in \mathcal{C}$, all minors of G also belong to $\mathcal{C}$. One of the deepest results in graph theory is the seminal Robertson-Seymour Theorem [RS04] (also known as the Graph Minor Theorem). Proved over a series of 20 papers spanning decades, the theorem establishes that finite graphs form a *well-quasi-ordering* under the minor relation.

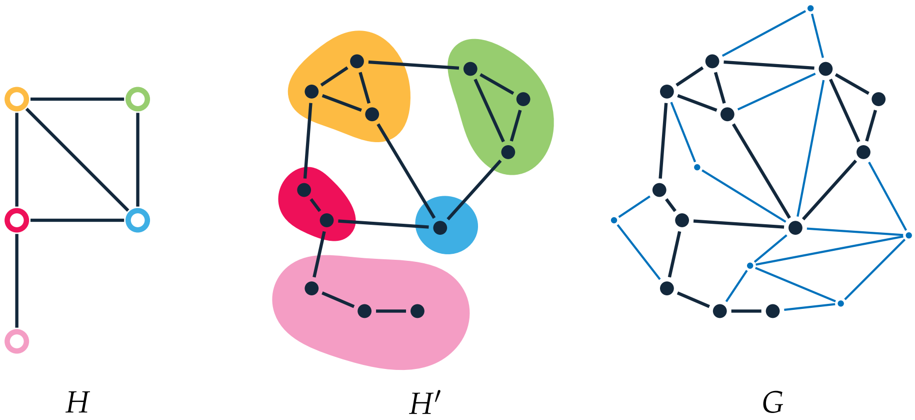


Figure I.8: An illustration where H is a minor of G, with its model H'.

A corollary of this theorem is that every minor-closed class of graphs can be completely characterized by a finite list of *forbidden minors*. If a graph does not contain any of the forbidden minors from this finite list, it is guaranteed to belong to the class. Minor-free graph classes often studied in discrete mathematics and possess many highly desirable algorithmic and structural properties. Examples of minor-free classes include the following.

- Forests, the class of acyclic graphs is closed under minors and is entirely characterized by excluding the triangle K_3 as a minor.

- Series parallel graphs are the graphs obtained by starting from a pair of terminal vertices and applying recursively two operations, series and parallel composition (see Figure I.9). Those graphs are exactly the ones that exclude K_4 as a minor.

- Planar Graphs are, by Wagner's Theorem [Wag37], exactly the graphs that do not contain the complete graph K_5 or the complete bipartite graph $K_{3,3}$ as a minor.

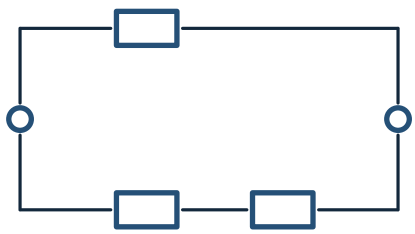


Figure I.9: The operations of series parallel graphs are defined just like when connecting two electric circuits in series or parallel.

Graph minors are interesting from an algorithmic point of view. For example, many NP-hard problems admit polynomial-time algorithms or approximation schemes when the input graphs are restricted to a class excluding a fixed minor. Beyond algorithmic considerations, the minor relation is profoundly interesting because it captures the fundamental, macroscopic topology of a network. While induced subgraphs are highly sensitive to local changes, such as subdividing a single edge, minors abstract this away and reveal the underlying skeleton of the graph, creating a bridge to topological geometry. For instance, the forbidden minors for planar graphs (K_5 and $K_{3,3}$) are the exact topological obstructions to embedding a graph on a two-dimensional sphere without edge crossings. The broader Graph Minor Structure Theorem generalize this idea, proving that any minor-closed class can essentially be decomposed into pieces that are "almost" embeddable on surfaces of bounded genus.

I.2 Distributed algorithms

This section gives a brief overview of the field of distributed algorithms, and particularly surrounding the LOCAL model. If you are already familiar with those notions, feel free to skip this section.

Background. The field of distributed algorithms studies algorithms designed to run simultaneously on multiple processors without shared memory. These algorithms often govern the behavior of computer networks, such as the Internet. To study such networks, we represent them as graphs, where vertices correspond to computers and edges to communication links between these computers.

Many different models have been introduced to describe the behavior of these networks. Some are highly abstract, making them easy to analyze but often too unrealistic for practical applications. Conversely, other models are much lower-level: they aim to capture real-world characteristics such as unreliable communications, non-unique identifiers, and other implementation details. While these models are more realistic, they are also much harder to work with and often too complex to give clear and meaningful results.

The LOCAL model. The LOCAL model offers a middle ground between these two extremes. It is simple enough to allow the study of general problems while still capturing essential aspects of real networks. This model, popularized by Linial in his seminal papers [Lin87, Lin92], has been primarily used to study the locality of problems in computer networks. That is, to understand which problems can be solved when each vertex only has access to information from a restricted neighborhood around itself.

In the LOCAL model, a network is represented by a simple connected graph. Each vertex $v \in V(G)$ represents a computer with potentially unlimited computational power and can only communicate with its neighbors $N(v) := \{u \mid uv \in E(G)\}$. Initially, a vertex v only knows its own unique identifier. See Section I.2 for an illustration. The same algorithm is executed independently on each vertex, so there is no shared memory between them.

To gather information, vertices communicate in synchronous rounds. At the beginning of each round, each vertex can send a message of arbitrary size to each of its neighbors. At the end of the round, it receives the messages sent by its neighbors and performs local computations. These computations determine the messages that will be sent in subsequent rounds. Because there are no time or memory restrictions on local computation, a vertex can perform any computable operation. A vertex can also decide to terminate its execution, output a value, and halt. The algorithm terminates once all vertices have halted.

The global output of the algorithm is derived from the set of outputs

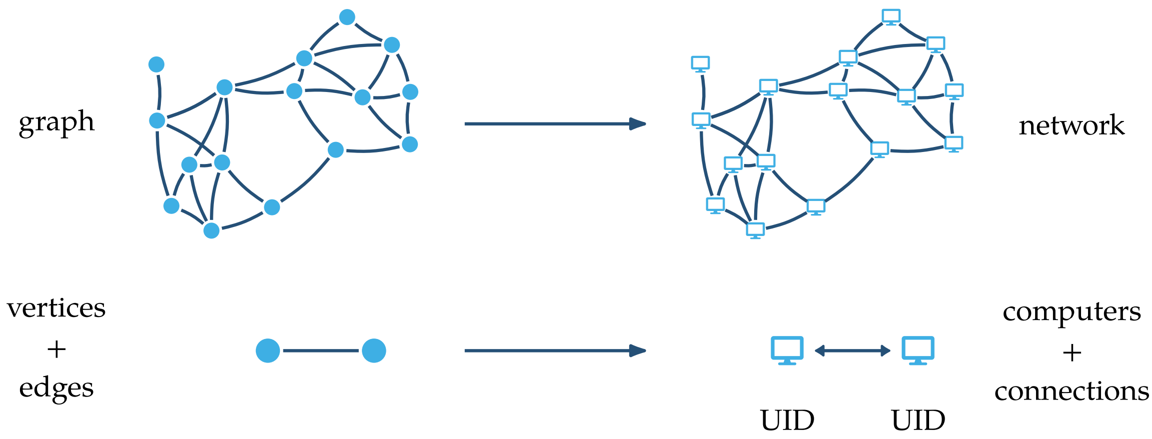


Figure I.10: The relationship between a network and its graph.

produced by all vertices once the algorithm terminates. In many graph problems, this means each vertex indicates whether or not it belongs to the desired solution.

There is another equivalent definition of r-round local algorithms. Instead of communicating in rounds, a vertex v can simply gather all the edges and vertices it sees in its r-round view and then compute its final state locally in a single step. The r-round view of v is composed of the vertices at distance at most r from v (along with the identifiers of those vertices) and all the edges with one endpoint at distance $r-1$ of v. Said differently, it contains all vertices and edges that are in any path of length at most r starting from v. For illustrations, see Section I.2. Formally, executing a distributed algorithm in r rounds is equivalent to evaluating a function f that maps these local neighborhood with identifiers directly to the output space.

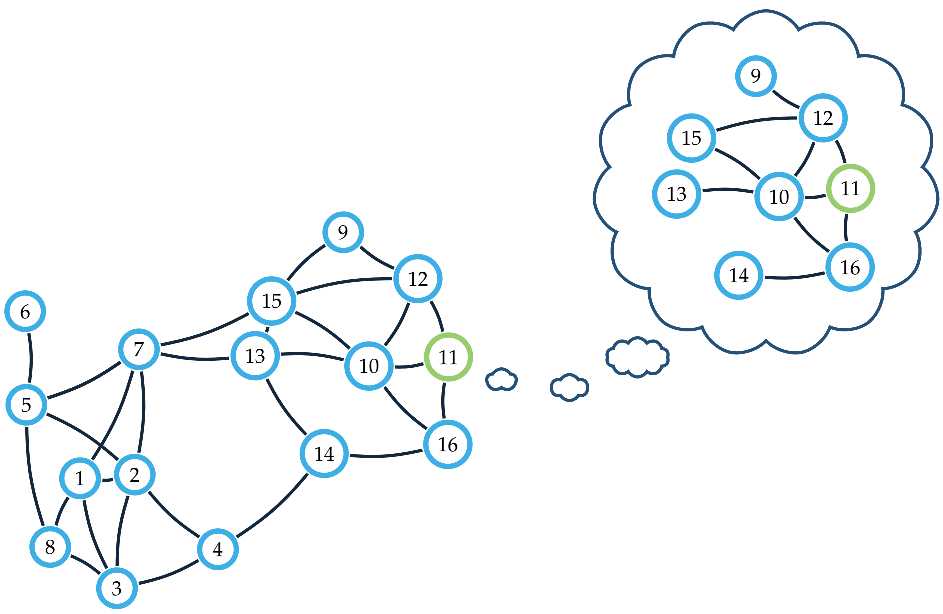


Figure I.11: A vertex of identifier 11 along with its local view at distance 2.

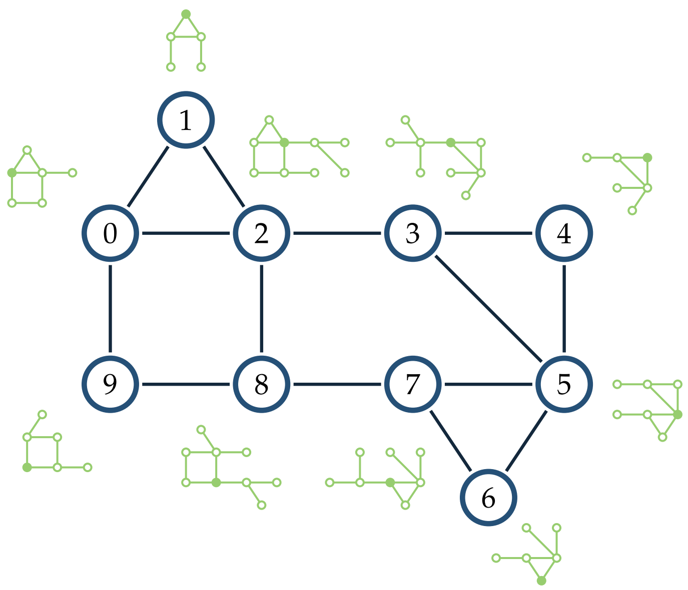


Figure I.12: A graph G along with all the local views (in green) of the vertices after 2 rounds. This figure is taken from Cyril Gavoille's lecture notes [Gav25] (with authorization).

An example: dominating sets. For example, consider the Minimum Dominating Set (MDS) problem. A set $D \subseteq V(G)$ is a dominating set if every vertex $v \in V(G)$ either belongs to D or has a neighbor in D. If, furthermore, D has minimum cardinality, it is a minimum dominating set. Its size is denoted $\mathrm{MDS}(G)$.

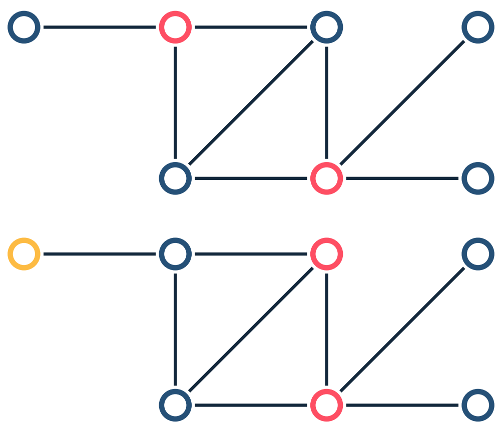


Figure I.13: At the top, the set of red vertices forms a dominating set. This is not the case for the bottom graph, as the orange vertex is not dominated.

In the MDS problem, each vertex outputs either "*yes*" or "*no*", and the set Y of vertices outputting "*yes*" constitutes the solution. For the algorithm to correctly solve the problem, Y must be a dominating set of minimum cardinality.

Algorithm complexity. The complexity of a problem in the LOCAL model is measured by the number of communication rounds required before every vertex has produced its output. On a connected graph, any problem whose solution is computable can be solved in at most $n = |V(G)|$ rounds. Indeed, in each round, every vertex can send its neighbors all the information it currently possesses. Because the graph is connected, after n rounds, every vertex will have gathered enough information to reconstruct the entire graph. Since local computation is unrestricted, each vertex can then compute a solution. This is why the primary focus lies on local algorithms whose running time is sublinear in the number of vertices, and even the diameter.

From a broader perspective, LOCAL algorithms act as local proofs of global graph properties, offering an information-theoretic view of distributed computing. They allow us to precisely quantify how much local information is necessary to solve a given problem by establishing both positive algorithmic results and negative lower bounds. A classic example is planar graph coloring. The famous Four Color Theorem [AH89, RSST96, IKM+26] guarantees that every planar graph admits a proper 4-coloring. However, it is impossible to compute such a 4-coloring in $o(|V(G)|)$ rounds in the LOCAL model [ABBE18]. This lower bound reveals that to successfully 4-color a single vertex of a planar graph, one must examine a linear-sized portion of the entire network.

Separating the LOCAL model from the centralized model. The problem of finding a minimum dominating set is a classical NP-complete problem and has been intensively studied for its many practical and theoretic uses. It is also hard to approximate within a factor $\log \Delta(G)$ [CC08] in polynomial time, where $\Delta(G)$ is the maximum degree of G (under the assumption that $\mathsf{P} \neq \mathsf{NP}$).

It is important to differentiate the difficulty of solving a problem in the centralized model and in the LOCAL model. To highlight this, we will look at two different examples: one problem that is intractable centrally but trivial locally, and another that is trivial centrally but strictly impossible to solve locally.

Consider the problem of solving Minimum Dominating Set on a specific family of graphs $\mathcal{C}$. We construct this family as follows: for any arbitrary graph G, create a new graph G' by adding some new vertex c, and a new vertex v' for each $v \in V(G)$. Add the edges cv' and vv' for each $v \in V(G)$ to form the new graph G' as in Figure I.2. The resulting family $\mathcal{C}$ consists of all such graphs G'.

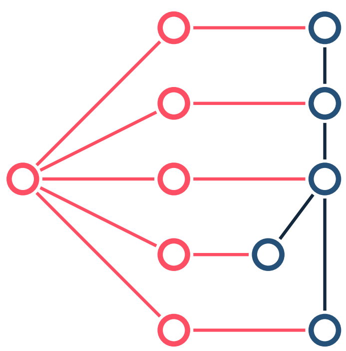


Figure I.14: In black, G, and in red, the new added edges and vertices to form G'

By construction, the diameter of any graph $G' \in \mathcal{C}$ is at most 4. Therefore, in the LOCAL model, every vertex can gather the entire topology of G' in just 4 communication rounds. Because local computation is unbounded, each vertex can then deterministically compute the same exact[3] MDS of G' and output its own participation in the set. Thus, MDS on $\mathcal{C}$ can be solved exactly in 4 rounds. Therefore, on this class of

[3] For instance, by computing all possible ones and choosing the lexicographically smallest one.

graphs, Minimum Dominating Set is easy to solve locally.

However, solving MDS on G' in the standard centralized setting is NP-hard. As long as G contains at least one edge, there always exists an optimal dominating set of G' that includes c and the exact MDS of G. Consequently, finding the size of the MDS of G' is equivalent to finding the MDS of the (arbitrary) graph G. For that reason, it is possible that a problem is hard in the centralized model but easy in the LOCAL model.

The converse is also true: a problem can be very easy to solve in the centralized model where we know the whole graph, and hard in the LOCAL model (requiring $\Omega(n)$ rounds). Consider the problem of 2-coloring a path P_n. In the centralized model, this is trivial: one simply alternates colors 1 and 2 along the vertices of the path.

Figure I.15: A 2-colored path P_{10}.

However, in the LOCAL model, it is not possible to 2-color paths in $o(n)$ rounds, meaning the problem is very hard to solve. To see why, assume for the sake of contradiction that such a local algorithm exists. We could then execute this exact algorithm on an even path P_{2n} and on an odd path P_{2n+1}. Because the algorithm runs in less than $n/2$ rounds, one can assign identifiers to P_{2n} and P_{2n+1} so that the view of any endpoint vertex is the same in both paths. The algorithm would therefore 2-color both paths with the same color assignment on the ends. This yields a contradiction, so paths not 2-colorable in the LOCAL model in less than $n/2$ rounds.

For similar locality-based reasons, computing an *exact* minimum dominating set in the LOCAL model cannot be done in $O(1)$ rounds for general graphs. Consequently, distributed research heavily focuses on finding efficient approximations. A *k-approximation algorithm* for a minimization problem returns an output set D that satisfies all problem constraints (e.g., being a valid dominating set) while having its size bounded by $k|D^*|$, where D^* is the optimal solution.

However, even obtaining a constant-factor approximation for MDS maybe be hard for some graphs. It has been proven that any r-round local algorithm on n-vertex graphs with maximum degree Δ can only approximate the minimum dominating set within a factor of $\Omega(n^{\frac{1-o(1)}{4r^2}}/r)$ and $\Omega(\Delta^{1/(r+1)}/r)$ [KMW16] of the optimal solution. Consequently, achieving any general constant-factor approximation requires at least $\Omega(\sqrt{\log n/\log\log n})$ and $\Omega(\log\Delta/\log\log\Delta)$ rounds.

Because of this reason, we must restrict our attention to more specific classes of graphs.

I.3 *Main results of this thesis*

This thesis explores the landscape of distributed graph algorithms, focusing on the relationship between network topology, time complexity, and approximation quality in the LOCAL model. The main part of this work is identifying the structural properties of graphs required to make distributed problems either tractable or hard. We begin by studying exact solutions in highly structured environments like trees, before transitioning to approximate solutions in increasingly complex graph classes. Along the way, we examine how topological and graph-theoretic parameters, such as asymptotic dimension and pathwidth, dictate the limits of distributed computation.

This thesis is divided into 4 main parts.

A first taste of distributed algorithms. Part A, which you are currently reading, is composed of 2 chapters; Chapter I serves as an introduction, and Chapter II delves deeper in the field of distributed algorithms. In more detail, this particular chapter studies binary labeling problems, where the goal is to select a subset of edges in a 2-colored tree such that some degree constraints on the incident edges of white and black nodes are satisfied. Prior work [BBE+20] classified the complexity of these problems on bounded degree graphs into three categories: $\Theta(1)$, $\Theta(\log n)$ or $\Theta(n)$. We refine this by analyzing the complexity as a function of three parameters: n, the degree of white nodes d, and the degree of black nodes δ. We only focus on structurally simple problems, which captures for example the ones describable by a context-free grammar. The intuition here is that high-degree nodes fundamentally alter the flow of information in a distributed setting. To take advantage of this, we develop a more aggressive variant of the rake-and-compress technique. While traditional rake-and-compress iteratively removes leaves and shrinks paths, our adaptation specifically uses high-degree nodes to accelerate the compression of the tree. This allows us to classify every structurally simple problem into one of the five categories: $\Theta(1)$, $\Theta(\log_d n)$, $\Theta(\log_\delta n)$, $\Theta(\log n)$ or $\Theta(n)$. This work has been published in SIROCCO'24 [LPS24].

Approximating in the LOCAL model. Part B focuses on constant-factor approximations problems such as Minimum Dominating Set (MDS), on broader graph classes. In Chapter III, we prove that given an α-approximation LOCAL algorithm for MDS on planar graphs, we can construct a $(3\alpha + 1)$-approximation LOCAL algorithm running in $f(g)$ rounds for MDS on graphs embeddable in a surface of Euler genus g. Building upon the $(11 + \varepsilon)$-approximation algorithm by Heydt et al. [HKO+25], we deduce a $(34 + \varepsilon)$-approximation ratio for graphs of genus g. This result significantly improves upon the previous state-of-the-art factor of $24g + O(1)$ established by Amiri et al. [ASS19], as well as the $91 + \varepsilon$ factor obtained by Czygrinow et al. [CHWW19] in the specific case of orientable surfaces.

We then generalize this result in two directions. First, we consider other graph problems studied in distributed algorithms, such as Minimum k-Tuple Dominating Set, for which constant-time approximation algorithms were known for planar graphs, but not for bounded-genus graphs. Second, we extend our results to graph classes beyond bounded genus. We establish these results through a series of meta-theorems concerning certain kinds of minimization problems (called *cuttable* problems) that admit constant-round LOCAL approximation algorithms. Intuitively, for cuttable problems, it is possible to systematically extract small subgraphs whose local solutions are proportional to the global solution restricted to the subgraph's neighborhood. This work has been accepted at PODC'26.

Next, Chapter IV shows that certain structured graphs (those excluding a minor $K_{2,t}$) admit a deterministic distributed algorithm in $f(t)$ rounds that computes a 50-approximation for the Minimum Dominating Set problem. Although fast, approximate distributed algorithms for these problems were already known for H-minor-free graphs, all previously exhibited an approximation ratio that depended on H. This work appears in PODC'25 [BGPW25].

A key new ingredient in the analysis of the various distributed algorithms of Chapters III and IV is the use of *asymptotic dimension*, a geometric concept introduced by Gromov in 1993 [Gro93]. Broadly, this allows us to prove global-to-local arguments, up to a multiplicative overhead on the bounds of the statements.

Going further, we conjecture in Chapter V that the bottleneck for distributed approximation is not the size of the excluded minor, but its pathwidth. We provide a matching lower bound to show that pathwidth would be the optimal parameter.

Exploring pathwidth. To prove our conjecture, we must fundamentally understand pathwidth, which is the focus of Part C. After a quick introduction to pathwidth in Chapter VI, the next Chapter VII addresses a purely structural graph theory problem: bounding the pathwidth of graphs whose edge sets can be covered by k shortest paths. We prove a polynomial bound of $O(k^4)$, improving on the best known bound of $O(3^k)$ [DFPT24]. The intuition here is that shortest paths impose a strict metric structure that naturally limits how complex the graph can be. For the case of $k \leqslant 3$, we show the best possible bound for the pathwidth is exactly k. This work has been published in STACS'26 [BMG+26].

Armed with these new structural insights, we return in Chapter VIII to tackle our pathwidth conjecture about approximating MDS in the distributed setting. By building on the techniques developed for $K_{2,t}$-minor-free graphs, we design a 50-approximation algorithm for graphs excluding any minor H of pathwidth at most 2. This is done by analyzing in more detail graphs that are locally well-connected everywhere and that have high diameter. Notably, this confirms our conjecture for the particular case $k = 2$. This work has not been published yet.

Expanding the horizon. Part D, the final part of the thesis, maps out future research directions and ideas, and possible open questions. In Chapter IX, we discuss the expectations and necessary theoretical bridges to extend our algorithms to graphs forbidding minors of pathwidth k for arbitrary k. In Chapter X, we explore how our ideas can be used design constant-factor approximations for other classic problems such as Minimum Vertex Cover or Minimum Distance-k Dominating Set. Both those chapters have not been published yet. Finally, Chapter XI provides a short conclusion of the thesis along with a summary of some questions left open by our works.

I.4 *Works not included in this thesis*

In addition to the works presented here in this thesis, I have also worked on a number of different topics. Here is a list of the published ones, by chronological order, from oldest to newest. During research internships, I participated in three different projects that lead to publications:

- With Ngoc-Trung Nguyen and Binh-Minh Bui-Xuan, I studied matching problems in temporal graphs [PNB21]. In this problem called γ-*Matching*, the goal is to choose the maximum amount of pair working sessions of duration at least γ. We design exponential exact algorithms and polynomial time approximation schemes when the data has a bounded "density". This work appears in CIAC'21.
- With Mathieu Mari and Michał Pilipczuk, I studied parameterized approximations algorithms for a generalization of the Knapsack problem to two dimensions [MPP23], where we are given a set of axis-parallel rectangles and a rectangular bounding box, and the goal is to pack as many of these rectangles inside the box without overlap. We give parameterized approximation schemes when the input rectangles are wide (its width is greater that its height). This work has been published in IPEC'23.
- With Sameep Dahal, Francesco d'Amore, Henrik Lievonen, and Jukka Suomela, I worked on the derandomization of generic algorithms in the LOCAL model [DdL+23]. In this work, we revisit a derandomization result of Chang, Kopelowitz, and Pettie [CKP19] that states that any randomized LOCAL algorithm locally solving a checkable labeling problem (LCL) can be derandomized with at most exponential overhead, if the number of random bits is bounded by the input size. We give a new proof that removes this unnecessary assumption. This work appears in DISC'23.

During my PhD, I also worked with wonderful people.

- As part of a research visit with Édouard Bonnet, we worked with Pierre Aboulker and Nicolas Trotignon on the problem of finding induced minors in graphs [ABPT25], which are a variant of graph minors where one is not allowed to remove edges. We prove that the problem of Induced 2-Disjoint Paths is NP-complete on string graphs (intersection graphs of curves in the plane). This has applications for several induced minor containment problems, answering questions of Korhonen and Lokshtanov [KL24], Chudnovsky, Seymour, and Trotignon [CST13] and Le [Le19]. This work has been published in ICALP'25.

- While co-advising the internship of Théotime Leclere with Marthe Bonamy, we worked on variant of the Turán number [BLP25]. We solve a recent question of Caro, Patkós and Tuza [CPT25] by determining the exact maximum number of edges in a bipartite connected graph as a function of the longest path it contains as a subgraph and of the number of vertices in each side of the bipartition. This work has been submitted to a journal.
- With Samuel Humeau, Mamadou Moustapha Kanté, Daniel Mock, and Alexandre Vigny, we studied the problem of subgraph freeness in property testing [HKM+26]. We prove that for any class $\mathscr{C}$ with bounded expansion[4] and any graph H, deciding whether G is far or not from excluding a subgraph H can be done with a constant number (depending on H and $\mathscr{C}$ but not G) of queries, for any $G \in \mathscr{C}$. This work appears in STACS'26.
- While participating in the private 2024 RW-DIST workshop in l'Aquila, we worked with Alkida Balliu, Sebastian Brandt, Dennis Olivetti, Fabian Kuhn, and Gustav Schmid on how the knowledge of the network size impacts the complexities of LOCAL algorithms in trees. Among other results, we prove that for some type of problems, there is an infinite amount of discrete complexity regions that exist. While other landscape theorems[5] have complexities of the form $\Theta(n^{1/k})$, in our setting, the exponents become very bizarre. This work has been accepted in PODC'26.

[4] A graph class has bounded expansion if, for every integer r, all its r-shallow minors (minors where branch sets have radius at most r) have average degree bounded by a function of r only. A lot of sparse graphs classes, such as minor-closed classes, have this property.

[5] For more information, see Section II.1.1.

References

[ABBE18] Pierre Aboulker et al. "Distributed coloring in sparse graphs with fewer colors". In: *ACM Symposium on Principles of Distributed Computing (PODC 2018)*. Egham, United Kingdom, July 2018, pp. 419–425. DOI: 10.1145/3212734.3212740. URL: https://hal.archives-ouvertes.fr/hal-01710649.

[ABPT25] Pierre Aboulker et al. "Induced Disjoint Paths Without an Induced Minor". In: *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*. Ed. by Keren Censor-Hillel et al. Vol. 334. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, 4:1–4:14. ISBN: 978-3-95977-372-0. DOI: 10.4230/LIPIcs.

ICALP.2025.4. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2025.4.

[AH89] Kenneth I Appel and Wolfgang Haken. *Every planar map is four colorable*. Vol. 98. American Mathematical Soc., 1989.

[ASS19] Saeed Akhoondian Amiri, Stefan Schmid, and Sebastian Siebertz. "Distributed Dominating Set Approximations beyond Planar Graphs". In: *ACM Transactions on Algorithms* 15.3 (2019), Article No. 39, pp. 1–18. DOI: 10.1145/3093239.

[BBE+20] Alkida Balliu et al. "Classification of Distributed Binary Labeling Problems". In: *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*. Ed. by Hagit Attiya. Vol. 179. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 17:1–17:17. DOI: 10.4230/LIPIcs.DISC.2020.17.

[BGPW25] Marthe Bonamy et al. "Local Constant Approximation for Dominating Set on Graphs Excluding Large Minors". In: *44th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. Huatulco, Mexico: ACM Press, June 2025. DOI: 10.1145/3732772.3733531.

[BLP25] Marthe Bonamy, Théotime Leclere, and Timothé Picavet. "Bipartite Turán number of paths and other trees". In: *arXiv preprint arXiv:2511.07374* (2025).

[BMG+26] Julien Baste et al. "A Polynomial Bound on the Pathwidth of Graphs Edge-Coverable by k Shortest Paths". In: *43rd International Symposium on Theoretical Aspects of Computer Science, STACS 2026, Grenoble, France, March 9-13, 2026*. Ed. by Meena Mahajan et al. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026, 10:1–10:17. DOI: 10.4230/LIPICS.STACS.2026.10. URL: https://doi.org/10.4230/LIPIcs.STACS.2026.10.

[CC08] M. Chlebík and J. Chlebíková. "Approximation hardness of dominating set problems in bounded degree graphs". In: *Information and Computation* 206.11 (2008), pp. 1264–1275. ISSN: 0890-5401. DOI: https://doi.org/10.1016/j.ic.2008.07.003. URL: https://www.sciencedirect.com/science/article/pii/S0890540108000928.

[CHWW19] Andrzej Czygrinow et al. "Distributed $\mathcal{CONGEST}_{BC}$ constant approximation of MDS in bounded genus graphs". In: *Theoretical Computer Science* 757 (Jan. 2019), pp. 1–10. DOI: 10.1016/j.tcs.2018.07.008.

[CKP19] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. "An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model". In: *SIAM*

Journal on Computing 48.1 (2019), pp. 122–143. DOI: 10.1137/17M1117537.

[CPT25] Yair Caro, Balázs Patkós, and Zsolt Tuza. "Bipartite Turán number of trees". In: *arXiv preprint arXiv:2502.09052* (2025).

[CST13] Maria Chudnovsky, Paul Seymour, and Nicolas Trotignon. "Detecting an induced net subdivision". In: *Journal of Combinatorial Theory, Series B* 103.5 (2013), pp. 630–641. ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2013.07.005. URL: https://www.sciencedirect.com/science/article/pii/S0095895613000531.

[DdL+23] Sameep Dahal et al. "Brief Announcement: Distributed Derandomization Revisited". In: *37th International Symposium on Distributed Computing (DISC 2023)*. Ed. by Rotem Oshman. Vol. 281. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 40:1–40:5. ISBN: 978-3-95977-301-0. DOI: 10.4230/LIPIcs.DISC.2023.40. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.DISC.2023.40.

[DFPT24] Maël Dumas et al. "On graphs coverable by k shortest paths". English. In: *SIAM Journal on Discrete Mathematics* 38.2 (2024), pp. 1840–1862. ISSN: 0895-4801. DOI: 10.1137/23M1564511.

[Gav25] Cyril Gavoille. *Algorithmique Distribuée*. Notes de cours. 2025.

[Gro93] Mikhael Gromov. *Geometric Group Theory: Asymptotic invariants of infinite groups*. Cambridge University Press, 1993.

[HKM+26] Samuel Humeau et al. "Testing H-Freeness on Sparse Graphs, the Case of Bounded Expansion". In: *43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026)*. Ed. by Meena Mahajan et al. Vol. 364. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026, 55:1–55:18. ISBN: 978-3-95977-412-3. DOI: 10.4230/LIPIcs.STACS.2026.55. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2026.55.

[HKO+25] Ozan Heydt et al. "Distributed domination on sparse graph classes". In: *European Journal of Combinatorics* 123 (Jan. 2025), p. 103773. DOI: 10.1016/j.ejc.2023.103773.

[IKM+26] Yuta Inoue et al. "The Four Color Theorem with Linearly Many Reducible Configurations and Near-Linear Time Coloring". In: *arXiv preprint arXiv:2603.24880* (2026).

[KL24] Tuukka Korhonen and Daniel Lokshtanov. "Induced-Minor-Free Graphs: Separator Theorem, Subexponential Algorithms, and Improved Hardness of Recognition". In: *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2024, pp. 5249–5275. DOI: 10.1137/1.9781611977912.188. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611977912.188. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611977912.188.

[KMW16] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. "Local Computation: Lower and Upper Bounds". In: *J. ACM* 63.2 (Mar. 2016). ISSN: 0004-5411. DOI: 10.1145/2742012. URL: https://doi.org/10.1145/2742012.

[Le19] Ngoc Khang Le. "Detecting an induced subdivision of K4". In: *Journal of Graph Theory* 90.2 (2019), pp. 160–171. DOI: https://doi.org/10.1002/jgt.22374. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jgt.22374. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.22374.

[Lin87] Nathan Linial. "Distributive graph algorithms – Global solutions from local data". In: *28th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, Oct. 1987, pp. 331–335. DOI: 10.1109/SFCS.1987.20.

[Lin92] Nathan Linial. "Locality in Distributed Graphs Algorithms". In: *SIAM Journal on Computing* 21.1 (1992), pp. 193–201. DOI: 10.1137/0221015.

[LPS24] Henrik Lievonen, Timothé Picavet, and Jukka Suomela. "Distributed Binary Labeling Problems in High-Degree Graphs". In: *Structural Information and Communication Complexity*. Ed. by Yuval Emek. Cham: Springer Nature Switzerland, 2024, pp. 402–419. ISBN: 978-3-031-60603-8.

[MPP23] Mathieu Mari, Timothé Picavet, and Michał Pilipczuk. "A Parameterized Approximation Scheme for the Geometric Knapsack Problem with Wide Items". In: *18th International Symposium on Parameterized and Exact Computation (IPEC 2023)*. Ed. by Neeldhara Misra and Magnus Wahlström. Vol. 285. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 33:1–33:20. ISBN: 978-3-95977-305-8. DOI: 10.4230/LIPIcs.IPEC.2023.33. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.IPEC.2023.33.

[PNB21] Timothe Picavet, Ngoc-Trung Nguyen, and Binh-Minh Bui-Xuan. "Temporal Matching on Geometric Graph Data". In: *Algorithms and Complexity*. Ed. by Tiziana Calamoneri and Federico Corò. Cham: Springer International Publishing, 2021, pp. 394–408. ISBN: 978-3-030-75242-2.

[RS04] Neil Robertson and P.D. Seymour. "Graph Minors. XX. Wagner's conjecture". In: *Journal of Combinatorial Theory, Series B* 92.2 (2004). Special Issue Dedicated to Professor W.T. Tutte, pp. 325–357. ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2004.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0095895604000784.

[RSST96] Neil Robertson et al. "Efficiently four-coloring planar graphs". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 571–575. ISBN: 0897917855. DOI: 10.1145/237814.238005. URL: https://doi.org/10.1145/237814.238005.

[Wag37] Klaus Wagner. "Über eine Eigenschaft der ebenen Komplexe". In: *Mathematische Annalen* 114.1 (1937), pp. 570–590.

Chapter II

Distributed binary labeling problems in high-degree trees

Abstract

Balliu et al. (DISC 2020) classified the hardness of solving *binary labeling problems* with distributed graph algorithms; in these problems the task is to select a subset of edges in a 2-colored tree in which white nodes of degree d and black nodes of degree δ have constraints on the number of selected incident edges. They showed that the deterministic round complexity of any such problem is $O_{d,\delta}(1)$, $\Theta_{d,\delta}(\log n)$, or $\Theta_{d,\delta}(n)$, or the problem is unsolvable. However, their classification only addresses complexity as a function of n; here $O_{d,\delta}$ hides constants that may depend on parameters d and δ.

In this chapter, we study the complexity of binary labeling problems as a function of all three parameters: n, d, and δ. To this end, we introduce the family of *structurally simple* problems, which includes, among others, all binary labeling problems in which cardinality constraints can be represented with a context-free grammar. We classify possible complexities of structurally simple problems. As our main result, we show that if the complexity of a problem falls in the broad class of $\Theta_{d,\delta}(\log n)$, then the complexity for each d and δ is always either $\Theta(\log_d n)$, $\Theta(\log_\delta n)$, or $\Theta(\log n)$.

To prove our upper bounds, we introduce a new, more aggressive version of the *rake-and-compress technique* that benefits from high-degree nodes.

Acknowledgements

This chapter is a version of a work with Henrik Lievonen and Jukka Suomela published in SIROCCO'24 [LPS24] adapted to the thesis.

Contents

II.1 Introduction

In this work we take the first steps towards characterizing possible distributed computational complexities of graph problems as a function of two parameters: the number of nodes and the maximum degree of the graph. We study so-called *binary labeling problems* [BBE+20a], previously only studied for constant-degree graphs, and we extend their classification so that it is parameterized also by the degrees of the nodes. To do that, we also introduce a new version of the *rake-and-compress technique* [MR85] that benefits from high-degree nodes.

II.1.1 Broader context

The key goal in the field of distributed graph algorithms is understanding how fast a given graph problem can be solved in a distributed setting. In this work we focus on the LOCAL model of distributed computing: in each round all nodes can exchange messages with each of their neighbors, and the running time of the algorithm is the number of communication rounds until all nodes stop and announce their own part of the solution (say, their own color).

Landscape of distributed complexity. While a lot of early work in the field focused on individual problems, modern theory of distributed graph algorithms makes a heavy use of *structural* results that apply to a broad family of graph problems. The best-known example is *locally checkable labeling problems* (LCLs), first introduced by Naor and Stockmeyer [NS95]. These are problems in which the task is to label nodes or edges with labels from some finite set, subject to some local constraints—examples of such problems include vertex coloring, edge coloring, maximal independent set, and maximal matching.

By now, we have a very good understanding of the landscape of possible round complexities that *any* LCL problem may have, in settings like cycles, grids, trees, and general graphs [BHK+18, BBOS21, CP19, GHK18, BBOS20, FG17, RG20, BFH+16, CKP19, GS17]. For example, there is no LCL problem with a time complexity between $\omega(\log^* n)$ and $o(\log n)$ in the deterministic LOCAL model. Gap result like this are powerful tools in proving lower bounds and upper bounds. For example, as soon as we have an algorithm that solves some LCL problem in $o(\log n)$ rounds, we can immediately speed it up to $O(\log^* n)$ rounds for free.

However, there is one key limitation: the landscape of complexities is currently understood well only as a function of the number of nodes n, for a constant maximum degree $\Delta = O(1)$. For example, there are problems with complexities of the form $\Theta(\log_\Delta n)$, and problems with complexities of the form $\Theta(\log n)$, but we do not have a more fine-grained understanding of all possible complexity classes as a function of both n and Δ.

Beyond constant degrees: fundamental challenges. LCLs as they were originally introduced by Naor and Stockmeyer [NS95] only pertain to graphs of some constant maximum degree. To generalize beyond constant Δ, we need a meaningful definition of the problem family.

Unfortunately, many natural generalizations lead to uninteresting results. If the local constraints may depend on the degrees in arbitrarily complicated ways, we can construct artificial problems with complexities of the form $\Theta(\log_{f(\Delta)} n)$ for virtually any function $f(\Delta) = O(\Delta)$. For example, we could start with a problem that has complexity $\Theta(\log_\Delta n)$, and modify the problem definition so that nodes of degree d will ignore up to $d - f(d)$ adjacent leaf nodes; in essence, we turn nodes of degree d into nodes of degree $f(d)$, and adjust the complexity accordingly. However, this way we will learn nothing about the complexities that *natural* graph problems might have. Very recently, an interesting generalization of LCLs was studied in a paper of Schmid [Sch26]. It focuses on the polynomial regime, i.e. complexities of the form $\Theta(n^{1/k})$. We take a different approach and focus on logarithmic complexities (and lower).

Beyond constant degrees: our take. In this work we study the family of *binary labeling problems*, as defined by Balliu et al. [BBE+20a] (see Section II.1.2). These are a special case of LCLs; what is particularly attractive is that the complexity of any given binary labeling problem (in the deterministic LOCAL model) can be automatically deduced, and in the constant-degree case there are only four complexity classes in trees.

Our plan is to see exactly how the characterization of binary labeling problems can be generalized in a meaningful manner beyond constant degrees; the hope here is that this will also guide us when we seek to find the right definitions for generalizing results on all LCLs.

II.1.2 Binary labeling problems

Let us recall the key definitions from [BBE+20a]. In a binary labeling problem Π, the task is to choose a subset of edges $X \subseteq E$ in a tree $G = (V, E)$, subject to local constraints. The problem is defined as a tuple $\Pi = (d, \delta, W, B)$, where $d \in \{2, 3, \dots\}$ is the *white degree*, $\delta \in \{2, 3, \dots\}$ is the *black degree*, $W \subseteq \{0, 1, \dots, d\}$ is the *white constraint* and $B \subseteq \{0, 1, \dots, \delta\}$ is the *black constraint*. We assume that the input graph is properly 2-colored with colors white and black. We define that $X \subseteq E$ is a solution to problem Π if the following holds:

- If v is a white node of degree d, and v is incident to k edges in X, then $k \in W$.
- If v is a black node of degree δ, and v is incident to k edges in X, then $k \in B$.

That is, we only care about the labeling incident to white nodes of degree d and black nodes of degree δ; these nodes are called *relevant nodes*. It is often useful to imagine that white nodes represent "nodes" and black nodes represent "edges" (if $\delta = 2$) or "hyperedges" (if $\delta > 2$). We will usually assume that $\delta \leqslant d$ (otherwise we can exchange the roles of black and white nodes).

Examples. Here are a couple of examples of binary labeling problems (adapted from [BBE+20a]):

1. *Bipartite splitting:* $W = \{1, 2, \dots, d-1\}$ and $B = \{1, 2, \dots, \delta - 1\}$. Here the task is to split the set of edges in two classes: "red edges" X and "blue edges" $E \setminus X$, and all relevant nodes must be incident to at least one red and at least one blue edge.

2. *Bipartite matching:* $W = B = \{1\}$. If we interpret that each edge $\{u, v\} \in X$ indicates that u is matched with v, in this problem each relevant node must be matched with exactly one other node (that may or may not be relevant).

While the problems are well-defined in any 2-colored tree, it is often easiest to consider the case that the tree consists of only leaf nodes and relevant nodes. For example, bipartite matching is the task of finding a matching in which all internal nodes are matched.

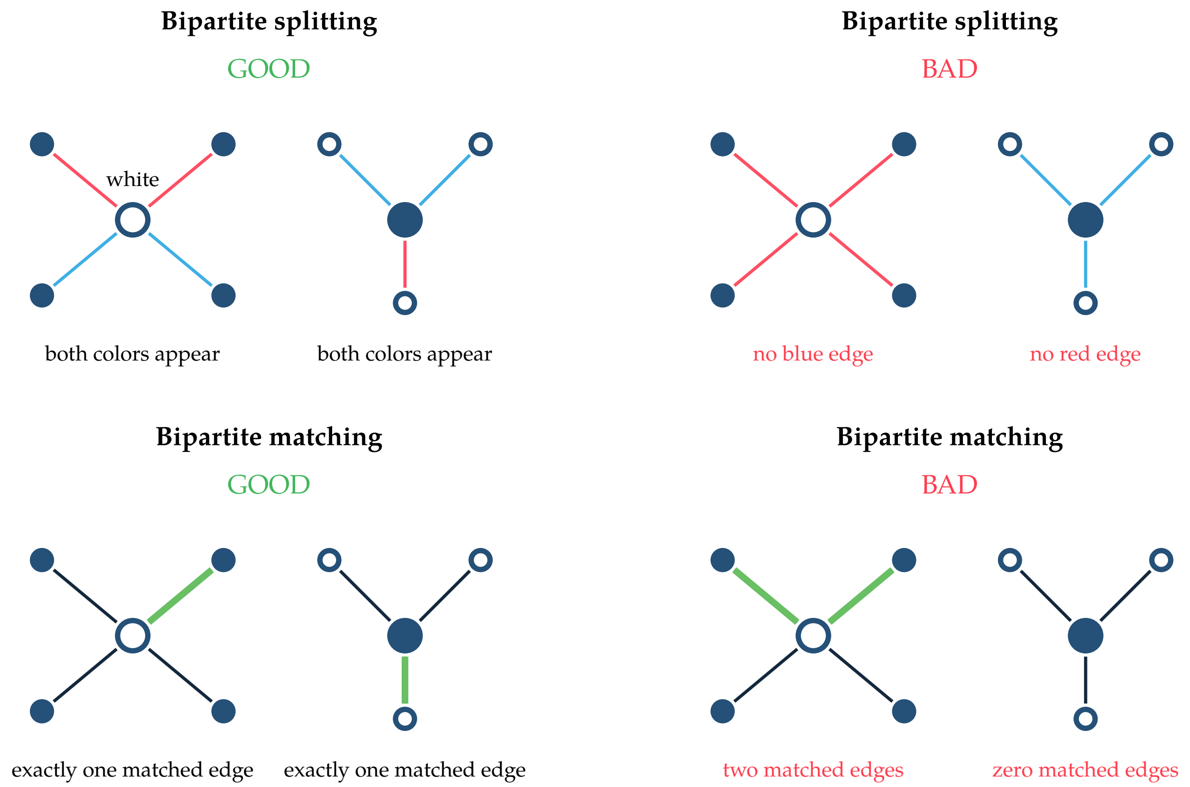


Figure II.1: Local good and bad configurations for bipartite splitting and bipartite matching.

Prior work. Given a tuple $\Pi = (d, \delta, W, B)$, we can directly look up the round complexity of Π in a table given by Balliu et al. [BBE+20a] (reproduced in Section II.1.2). For example, we immediately obtain:

1. *Bipartite splitting:* For $d = \delta = 2$ this problem requires $\Theta(n)$ rounds, but for any fixed constants $d > 2$, $\delta \geqslant 2$ the complexity is $\Theta_{d,\delta}(\log n)$.
2. *Bipartite matching:* For $d \geqslant \delta = 2$ this problem requires $\Theta_d(n)$ rounds, but for any fixed constants $d > 2$, $\delta > 2$ the complexity is $\Theta_{d,\delta}(\log n)$.

However, what their classification does not capture is the complexity as a function of d or δ; we have made this explicit above by writing $\Theta_{d,\delta}$, to emphasize that the hidden constants may depend on d and δ. For example, while there are numerous problems (such as the above two examples) with a complexity $\Theta_{d,\delta}(\log n)$, it is not at all clear what is the base of logarithm in each case. In particular, which binary labeling problems get easier when d and/or δ grows?

Deterministic complexity	**White constraint**	**Black constraint**
unsolvable	100^+	$0{*}{*}^+$
	00^+1	${*}{*}^+0$
	$0{*}{*}^+$	100^+
	${*}{*}^+0$	00^+1
	000^+	${*}{*}{*}^+$
	${*}{*}{*}^+$	000^+
$O_{d,\delta}(1)$	non-empty	111^+
	111^+	non-empty
	$1{*}{*}^+$	$1{*}{*}^+$
	${*}{*}^+1$	${*}{*}^+1$
$\Theta_{d,\delta}(n)$	10^+1	010
	010	10^+1
	$0^+1{*}$	${*}10^+$
	${*}10^+$	$0^+1{*}$
$\Theta_{d,\delta}(\log n)$	all other cases	

Table II.1: The complexity landscape from [BBE+20a].

II.1.3 Overview of contributions and new ideas

Main question. In this work we study parameterized families of binary labeling problems

$$\Pi(d, \delta) = (d, \delta, W(d), B(\delta)),$$

and our main question is this: what can we say about the complexity of any such problem family $\Pi(d, \delta)$, as a function of n, d, and δ?

Key technical challenge. We need to be careful not to make the family too broad—otherwise we will end up with an uninteresting result stating that there are artificial problems with virtually any complexity as a function of d and δ, without learning anything about natural graph problems.

Key results and new ideas. The main new insight is that we define the family of *structurally simple* problems. This definition captures how $W(d)$ and $B(\delta)$ can depend on d and δ. The aim is to exclude artificial pathological problems.

We then show that this definition is *useful* in the sense that we can prove strong statements about the complexity of structurally simple problems. For example, if the complexity as a function of n is $\Theta_{d,\delta}(\log n)$, then for each d and δ the complexity falls in one of these fine-grained classes: $\Theta(\log_d n)$, $\Theta(\log_\delta n)$, or $\Theta(\log n)$.

Finally, we show that the definition captures a *broad family* of problems: if the constraints $W(d)$ and $B(\delta)$ can be represented as binary strings in a context-free language, then $\Pi(d, \delta)$ is structurally simple. We will discuss this in more detail in Section II.1.4.

To prove our main result, we also needed to develop a new, more aggressive version of the rake-and-compress technique. We discuss this in more detail in Section II.1.5.

II.1.4 Contributions in more detail

Key definitions. In a family of binary labeling problems, both W and B are set families of the form $X(k) \subseteq \{0, 1, \ldots, k\}$. If we look at these set families in the examples given by Balliu et al. [BBE+20a, Table 3], we observe that they all fall in one of two classes, which we call *center-good* and *edge-good*. They are defined as follows.

A set $X \subseteq \{0, 1, \ldots, k\}$ is *ε-center-good* if there exists an $x \in X$ such that $k^\varepsilon \leqslant x \leqslant k^{1-\varepsilon}$. A set $X \subseteq \{0, 1, \ldots, k\}$ is *C-edge-good* if for all $x \in X$ we have $x \leqslant C$ or $x \geqslant k - C$.

Now we are ready to give the main definition. A set family $X(k) \subseteq \{0, 1, \ldots, k\}$ is *structurally simple* if there are some constants $0 < \varepsilon < 1$ and C such that $X(k)$ is either ε-center-good or C-edge-good for each k. A family of binary labeling problems $\Pi(d, \delta) = \big(d, \delta, W(d), B(\delta)\big)$ is *structurally simple* if both $W(d)$ and $B(\delta)$ are structurally simple.

For example, both the bipartite splitting problem and the bipartite matching problem are structurally simple, and so are all problems in [BBE+20a, Table 3]; in the bipartite splitting problem, $W(d)$ and $B(\delta)$ are center-good, while in the bipartite matching problem, $W(d)$ and $B(\delta)$ are edge-good. To give a pathological example of a set family that is not structurally simple, consider, for example, $X(k) = \{\lfloor \log k \rfloor\}$.

Main result. Our main result is a complete classification of structurally simple problems in the logarithmic region:

> **Theorem II.1.1.** *Let $\Pi(d, \delta)$ be a structurally simple family of binary labeling problems and assume that the complexity of $\Pi(d, \delta)$ in trees is $\Theta_{d,\delta}(\log n)$. Then the complexity of $\Pi(d, \delta)$ in trees for each d and δ falls in one of these classes, as long as $\delta \leqslant d = O(n^{1-\alpha})$ for some $\alpha > 0$:*
>
> *1.* $\Theta(\log n)$,

2. $\Theta(\log_\delta n)$,
3. $\Theta(\log_d n)$.

Remark II.1.2. *A few clarifying remarks are in order that help one to interpret the result:*

1. *Our result is constructive in the sense that we also show how to classify $\Pi(d, \delta)$ for any given d and δ (see Section II.4).*
2. *The complexity class may depend on d and δ. For example, we might have a problem in which for even values of d the complexity is $\Theta(\log_d n)$ and for odd values of d it is $\Theta(\log n)$.*
3. *Throughout this work, Θ-notation only hides constants that only depend on problem family Π, and not on d and δ; we write $\Theta_{d,\delta}$ explicitly if we are hiding constants that may depend on d and δ.*

Language-theoretic justification. A set $X \subseteq \{0, 1, \ldots, k\}$ can be also represented as a binary string $\hat{X}$ of length $k+1$: we index the bits with $i = 0, 1, \ldots, k$ and for each $i \in X$, the bit at index i in $\hat{X}$ is 1, and otherwise 0. Given a set family $X(k)$, we can then define a language of binary strings

$$\hat{X} = \{\hat{X}(2), \hat{X}(3), \hat{X}(4), \ldots\}.$$

Note that $\hat{X}$ is by construction *thin*: it contains at most one word of any given length [PS95].

Example II.1.3. *If $X(k) = \{1, 2, \ldots, k-1\}$, then $\hat{X}(k) = 01^{k-1}0$ and $\hat{X} = \{010, 0110, 01110, \ldots\} = 01^+0$. If $X(k) = \{1\}$, then $\hat{X}(k) = 010^{k-1}$ and $\hat{X} = \{010, 0100, 01000, \ldots\} = 010^+$. Here we use 1^ℓ to denote a sequence of ℓ 1s, and 1^+ to denote the sequence of one or more 1s.*

Equipped with this notation, we can study families of binary labeling problems from a language-theoretic perspective: any given problem family $\Pi(d, \delta)$ can be specified by giving a pair of languages $(\hat{W}, \hat{B})$. As long as these are languages over the binary alphabet, and there is exactly one string of each length $2, 3, \ldots$, such a pair of languages can be interpreted as a parameterized problem family $\Pi(d, \delta)$.

Many example problems in prior work [BBE+20a] correspond to *regular languages*. For example, bipartite splitting is $(01^+0, 01^+0)$, and bipartite matching is $(010^+, 010^+)$, while the *sinkless orientation problem* [BFH+16] is $(11^+0, 011^+)$. We take one step beyond regular languages and consider the case of *context-free languages*. The key observation is summarized in the following lemma:

Lemma II.1.4. *Let $X(k) \subseteq \{0, 1, \ldots, k\}$ be a family of sets. If $\hat{X}$ is a context-free language, then $X(k)$ is structurally simple.*

Corollary II.1.5. *Let $\Pi(d, \delta) = (d, \delta, W(d), B(\delta))$ be a family of binary labeling problems. If $\hat{W}$ and $\hat{B}$ are context-free languages, then $\Pi(d, \delta)$ is structurally simple.*

Note that the converse is not true.

This is the main justification for focusing on structurally simple problems: we have only excluded some pathological cases that are so complicated that they cannot be expressed with context-free grammars (this is also the reason why we call them structurally simple).

II.1.5 Key building block: a new rake-and-compress variant

A key tool for designing $O(\log n)$-time algorithms for binary labeling problems in trees has been the *rake-and-compress technique* [MR85]. A typical application of this technique proceeds as follows; we alternate between two steps:

1. Eliminate leaf nodes and isolated nodes.
2. Eliminate sufficiently long paths.

If we apply these steps for $O(\log n)$ times, in a tree of any shape, we will eliminate all nodes. Then we can construct a solution by working backwards: repeatedly put back one layer of nodes and construct a solution that makes these nodes happy; see [BBE+20a] for more detailed examples.

From our perspective, the main drawback of the rake-and-compress technique is that it does not benefit from high-degree nodes. For example, if we apply it in a tree in which all internal nodes have degree s, the worst-case complexity is still $\Theta(\log n)$, not $\Theta(\log_s n)$.

We introduce a new version of the technique that strictly benefits from high-degree nodes. Let us first rephrase the usual rake-and-compress procedure as follows (note that nodes in the middle of long paths are also nodes that are not close to higher-degree nodes):

1. Eliminate leaf nodes and isolated nodes.
2. Eliminate nodes that are not close to any node of degree 3 or more.

We generalize this as follows, so that we are much more aggressive with the compression step:

1. Eliminate leaf nodes and isolated nodes.
2. Eliminate nodes that are not close to any node of degree s or more.

We show that this leads to a procedure that completes in $O(\log_s n)$ rounds, and we show that we can use this more aggressive version of rake-and-compress to solve many binary labeling problems. This is the key building block that leads to the upper bounds $O(\log_d n)$ and $O(\log_\delta n)$ in Theorem II.1.1.

We believe that our new rake-and-compress technique will find applications also beyond binary labeling problems.

Broad class	Fine-grained classes	Reference
$O_{d,\delta}(1)$	$O(1)$	Proposition II.5.1
$\Theta_{d,\delta}(\log n)$	$\Theta(\log_d n)$ $\Theta(\log_\delta n)$ $\Theta(\log n)$	Theorem II.1.1
$\Theta_{d,\delta}(n)$	$\Theta(n/(d+\delta))$	Corollary II.5.6
unsolvable	unsolvable	

Table II.2: Summary of our results for structurally simple problems (for precise technical assumptions please refer to the theorem statements).

II.1.6 Questions for future work

While our classification is constructive, it also gives rise to a number of new open questions related to the *decidability* of complexities for entire problem families; we present here one example:

Question II.1.6. *Given context-free grammars for thin languages $\hat{W}$ and $\hat{B}$ that define a problem family $\Pi(d,\delta)$, how hard is it to decide if the complexity of $\Pi(d,\delta)$ is $\Theta(\log_d n)$ for all but finitely many values of d and δ?*

II.1.7 Roadmap

We first classify structurally simple problems in the logarithmic region:

- In Section II.2 we prove lower bounds of the forms $\Omega(\log n)$, $\Omega(\log_d n)$, and $\Omega(\log_\delta n)$.
- In Section II.3 we prove upper bounds of the forms $O(\log n)$, $O(\log_d n)$, and $O(\log_\delta n)$. Here we also introduce and use our new rake-and-compress technique.
- In Section II.4 we put together the lower and upper bounds and establish a full classification in the logarithmic region. This proves the main result, Theorem II.1.1.

Then in Section II.6 we show that all problems that can be defined with context-free grammars are indeed structurally simple; this establishes Lemma II.1.4 and Corollary II.1.5.

In Section II.5 we classify problems in the constant and linear complexity classes; this leads to a complete classification that we summarize in Section II.1.7.

II.2 Logarithmic lower bounds

We start by proving the main lower bound results. We establish the lower bounds of $\Omega(\log_d n)$, $\Omega(\log_\delta n)$, and $\Omega(\log n)$, depending on the structure of the problem.

II.2.1 A general lower bound

We start by showing a general lower bound that holds for any problem:

Lemma II.2.1. *Let $\Pi(d, \delta)$ be a family of binary labeling problems. For each d and δ the complexity of $\Pi(d, \delta)$ is either $O(1)$ or $\Omega(\log_d n)$, assuming $2 \leqslant \delta \leqslant d$.*

Proof. As proved by Balliu et al. [BBE⁺20a], all binary labeling problems that are not solvable with locality $O(1)$ have complexity $\Omega_{d,\delta}(\log n)$. They prove this by reducing all such problems to the *forbidden degree or sinkless orientation*, which is proved to be a fixed point of round elimination and not 0-round solvable.

It is known that there exists bipartite graphs with high girth. In particular, for any integers a, b, there exists (a, b)-biregular graphs with girth $\Theta(\log_{ab} n)$ [FLS⁺95]. By using techniques similar to Balliu et al. [BKK⁺23] on these graphs, we get an $\Omega(\log_d n)$ lower bound for Π. □

II.2.2 Definitions and problem transformations

To get more fine-grained lower bounds, we first define a few transformations for problems. With the help of these transformations, we can apply Lemma II.2.1 to prove stronger lower bounds.

We start by introducing the switch and the reverse of a problem and observe that they do not affect the complexity of the problem:

Let $\Pi(d, \delta) = (d, \delta, W(d), B(\delta))$ be a family of problems. The *switch* of $\Pi(d, \delta)$ is $\Pi^s(\delta, d) = (\delta, d, B(\delta), W(d))$.

Let $X(k) \subseteq \{0, \dots, k\}$. The *reverse* of $X(k)$ is $X^r(k) = \{k - x \mid x \in X\}$, and the *reverse* of $\Pi(d, \delta)$ is $\Pi^r(d, \delta) = (d, \delta, W^r(d), B^r(\delta))$.

Lemma II.2.2. *Binary labeling problem $\Pi^s(\delta, d)$ has the same complexity as $\Pi(d, \delta)$.*

Proof. Given an algorithm solving Π, we get an algorithm solving Π^s by reversing the role of white and black nodes. □

Lemma II.2.3. *Binary labeling problem $\Pi^r(d, \delta)$ has the same complexity as $\Pi(d, \delta)$.*

Proof. Given an algorithm solving Π, we get an algorithm solving Π^r by replacing the solution output with its complement. □

We now define a more complicated transformation called *shift* of a problem; see Section II.2.2 for an example. Let $X(d) \subseteq \{0, 1, \dots, d\}$. A *shift* by k of $X(d)$ is

$$X^{\leftarrow k}(d - k) = \{0, 1, \dots, d - k\} \cap \{x - i \mid x \in X(d), i \in \{0, 1, \dots, k\}\}.$$

$X(20)$:	·	·	·	·	·	5	·	·	·	·	·	·	·	·	·	15	·	·	·	·	·
$X^{\leftarrow 1}(19)$:	·	·	·	·	4	5	·	·	·	·	·	·	·	·	14	15	·	·	·	·	
$X^{\leftarrow 2}(18)$:	·	·	·	3	4	5	·	·	·	·	·	·	·	13	14	15	·	·	·		
$X^{\leftarrow 3}(17)$:	·	·	2	3	4	5	·	·	·	·	·	·	12	13	14	15	·	·			
$X^{\leftarrow 4}(16)$:	·	1	2	3	4	5	·	·	·	·	·	11	12	13	14	15	·				
$X^{\leftarrow 5}(15)$:	0	1	2	3	4	5	·	·	·	·	10	11	12	13	14	15					
$X^{\leftarrow 6}(14)$:	0	1	2	3	4	5	·	·	·	9	10	11	12	13	14						
$X^{\leftarrow 7}(13)$:	0	1	2	3	4	5	·	·	8	9	10	11	12	13							
$X^{\leftarrow 10}(10)$:	0	1	2	3	4	5	6	7	8	9	10										
$X^{\leftarrow 15}(5)$:	0	1	2	3	4	5															

Table II.3: Examples of the definition of a shift, assuming $X(20) = \{5, 15\}$.

Let $\Pi(d,\delta) = (d, \delta, W(d), B(\delta))$ be a family of problems. The *white shift* of Π is $\Pi^{\leftarrow_W k}(d,\delta) = (d, \delta, W^{\leftarrow k}(d), B(\delta))$, and the *black shift* of Π is $\Pi^{\leftarrow_B k}(d,\delta) = (d, \delta, W(d), B^{\leftarrow k}(\delta))$.

Lemma II.2.4. *Let $\Pi(d,\delta)$ be a family of problems with complexity $f(n,d,\delta)$. Then the white shift $\Pi^{\leftarrow_W k}$ has complexity $O(f((k+1)n, d+k, \delta))$ and the black shift $\Pi^{\leftarrow_B k}$ has complexity $O(f((k+1)n, d, \delta + k))$*

Proof. Let $\Pi(d,\delta) = (d, \delta, W(d), B(\delta))$ be family of binary labeling problems. We show that the white shift $\Pi^{\leftarrow_W k}$ has complexity $O(f((k+1)n, d+k, \delta))$; the proof for the black shift $\Pi^{\leftarrow_B k}$ follows by Lemma II.2.2.

Fix parameters d, δ and k. Let tree G be an instance for problem $\Pi^{\leftarrow_W k}$ with n nodes. We are only interested in black nodes of degree δ and white nodes of degree d as the rest of the nodes are unrestricted. Let G' be a copy of G with k additional black leaves attached to each white node of degree d. Note that G' has at most $n' = (k+1)n$ nodes. Let v be a white node of G, and let v' be the corresponding white node in G'. Denote the set of edges incident to v by $E_G(v)$, and edges incident to v' by $E_{G'}(v')$.

Suppose that X is a solution for $\Pi(d+k,\delta)$ on G'; such a solution can be computed from input graph G with locality $O(f((k+1)n, d+k, \delta))$ by assumption. Then X induces a solution $Y = X \cap E(G)$ for $\Pi^{\leftarrow_W k}$ on G. We know that $\deg_X(v') = \left|E_{G'}(v') \cap X\right| \in W(d+k)$. We now show that $\deg_Y(v) = \left|E_G(v) \cap Y\right| \in W^{\leftarrow k}(d)$.

It is clear that $0 \leqslant \deg_Y(v) \leqslant (d+k) - k = d$. Moreover,

$$\begin{aligned} \deg_Y(v) = \deg_X(v') - &\left|(E_{G'}(v') \setminus E_G(v)) \cap X\right|, \quad \text{and} \\ 0 \leqslant &\left|(E_{G'}(v') \setminus E_G(v)) \cap X\right| \leqslant \left|E_{G'}(v') \setminus E_G(v)\right| = k. \end{aligned}$$

Therefore, $\deg_X(v') - k \leqslant \deg_Y(v) \leqslant \deg_X(v')$. Combining these gives

$$\deg_Y(v) \in \{0, 1, \ldots, d\} \cap \left\{\deg_X(v') - k, \ldots, \deg_X(v')\right\} \subseteq W^{\leftarrow k}(d),$$

completing the proof. □

With the help of these transformations, we are now ready to prove lower bounds for problems. Indeed, if we can show that some problem

$\Pi^{\leftarrow_W k}(d,\delta)$ has complexity $\Omega(f(n,d,\delta))$, then we also know that Π has complexity $\Omega(f(n/(k+1), d-k, \delta))$.

II.2.3 Problem reductions

In this section, we use Lemmas II.2.2 to II.2.4 to reduce problems to easier ones and give lower bounds on those with the help of Lemma II.2.1. In what follows, we slightly abuse the notation and represent the sets $W(d)$ and $B(\delta)$ using the corresponding binary strings $\hat{W}(d)$ and $\hat{B}(\delta)$.

Proposition II.2.5. *Let k and l be integers. The problem $\Pi(d,\delta) = (d,\delta,0^{d-k+1}1^k, 1^l0^{\delta-l+1})$ for $d \geqslant k$, $\delta \geqslant l$ is either unsolvable or has deterministic complexity $\Omega(\log n)$ when $d,\delta = O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.*

Proof. Let $W = 0^{d-k+1}1^k$ and $B = 1^l0^{\delta-l+1}$. We reduce Π to $\Pi'(k,l) = (k,l,01^k,1^l0)$ by first applying white shift to get $\Pi_1 = \Pi^{\leftarrow_W d-k}$ and then black shift to get $\Pi' = \Pi_2 = \Pi_1^{\leftarrow_B \delta-l}$.

By Lemma II.2.1, $\Pi'(k,l)$ has complexity $\Omega(\log n)$. Indeed, k and l are constants, so one can verify this by consulting the classification table of Balliu et al. [BBE+20a]. Therefore, by Lemma II.2.4, Π_1 has complexity $\Omega(\log(n/(\delta-l+1)))$. Applying Lemma II.2.4 again, we get that Π has complexity

$$\Omega\left(\log\left(\frac{n}{(\delta-l+1)(d-k+1)}\right)\right) = \Omega(\log n)$$

when $d,\delta = O(n^{1-\varepsilon})$. □

Remark II.2.6. *Notice that if $k \leqslant 2$ and $l \leqslant 2$ then problem $\Pi(d,\delta) = (d,\delta,0^{d-k+1}1^k, 1^l0^{\delta-l+1})$ is either unsolvable or has complexity $\Theta(n)$.*

Proposition II.2.7. *Let k and l be integers. The problem $\Pi(d,\delta) = (d,\delta,1^k0^{d-k-l+1}1^l, 01^{\delta-1}0)$ for $d \geqslant k+l$ has deterministic complexity $\Omega(\log_\delta n)$ when $d = O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.*

Proof. We apply white shift to reduce $\Pi(d,\delta)$ to problem

$$\Pi'(k+l,\delta) = \Pi^{\leftarrow_W d-k-l} = (k+l,\delta,1^k01^l,01^{\delta-1}0).$$

Again, by Lemma II.2.1, problem Π' has complexity $\Omega(\log_\delta n)$. By previous results [BBE+20a], we know that Π' is neither a trivial nor an unsolvable problem. Applying Lemma II.2.4, we get that Π has complexity $\Omega(\log_\delta(n/(d-k-l+1))) = \Omega(\log_\delta n)$ when $d = O(n^{1-\varepsilon})$. □

II.3 Logarithmic upper bounds

Now we proceed to prove upper bounds; later in Section II.4 we will see that our lower and upper bounds indeed provide tight bounds for all structurally simple problems in the logarithmic region.

II.3.1 Center-good problems

Recall the concept of center-good constraints that we introduced in Section II.1.4. We start by showing that problems whose white or black constraint is center-good can be solved efficiently:

Lemma II.3.1. *Let $\Pi(d,\delta) = (d,\delta,W(d),B(\delta))$ be a solvable structurally simple problem. If $W(d)$ is center-good, then $\Pi(d,\delta)$ can be solved with locality $O(\log_d n)$. Similarly, if $B(\delta)$ is center-good, then $\Pi(d,\delta)$ can be solved with locality $O(\log_\delta n)$.*

To prove the lemma, we will first show that problems of this type are (s,t)*-resilient* [BBE+20a] for some integers s,t. A binary labeling problem $\Pi(d,\delta) = (d,\delta,W(d),B(\delta))$ is said to be (s,t)*-resilient* [BBE+20a] if:

- the string $\hat{W}(d)$ does not contain a substring of the form 0^{d+1-s}, and
- the string $\hat{B}(\delta)$ does not contain a substring of the form $0^{\delta+1-t}$.

Resilient problems are useful because they allow partial labelings (defined in the following) to be completed.

Let $G = (V,E)$ be a tree. We call $\ell\colon E' \to \{0,1\}$ for some subset of edges $E' \subseteq E$ a *partial labeling of* G. Given two partial labelings $\ell_1 : E_1 \to \{0,1\}$ and $\ell_2 : E_2 \to \{0,1\}$ on G, we say that ℓ_2 is a *completion* of ℓ_1 if $E_1 \subseteq E_2$, that is, it is defined on a larger set of edges, and if ℓ_1 and ℓ_2 agree on E_1. Given a problem Π and a partial labeling ℓ, we say a node $v \in V$ is *labelled in a valid manner* if the set of edges incident to v is in the domain of ℓ and if ℓ satisfies the constraints of Π on v.

Lemma II.3.2 (Balliu et al. [BBE+20b])**.** *Let $\Pi(d,\delta)$ be a (t,s)-resilient problem, let $G = (V,E)$ be a tree, and let ℓ be a partial labeling on G. There for every node $v \in V$, there exists a completion of ℓ that labels v in a valid manner if one of the following conditions holds:*

- *v is a white node incident to at most t edges with non-empty labels, or*
- *v is a black node incident to at most s edges with non-empty labels.*

Our plan is to recursively remove layers of nodes from the input tree. Then we can put the layers back and complete the layering by exploiting Lemma II.3.2. We first define a procedure DEG.

Let $G = (V_W \sqcup V_B, E)$ be a tree with bipartition (V_W, V_B), and

$$\mathrm{degen}_{s,t}(G) = \{v \in V_W \mid \deg_G(v) \leqslant s\} \cup \{v \in V_B \mid \deg_G(v) \leqslant t\}.$$

The procedure DEG(s,t) partitions all nodes of G into non-empty sets $L_1, L_2, \ldots, L_k = \varnothing$ for some k as follows:

$$\begin{aligned} G_0 &= G, \\ L_{i+1} &= \mathrm{degen}_{s,t}(G_i) \quad \text{if } G_i \text{ is not empty}, \\ G_{i+1} &= G_i \setminus L_{i+1}. \end{aligned}$$

We now show that computing decomposition $\mathrm{DEG}(s,t)$ can be computed with locality $O(\log_{\max\{s,t\}} n)$ in the LOCAL model. We start with the following lemma showing that the number of high-degree nodes in a tree is bounded:

Lemma II.3.3. *Let $d \geqslant 1$ and let $G = (V_W \sqcup V_B, E)$ be a tree with bipartition (V_W, V_B), where V_W is the set of white nodes and V_B is the set of black nodes. There are at most $(|V_B| - 1)/(d-1)$ white nodes with degree at least d, and at most $(|V_W| - 1)/(d-1)$ black nodes with degree at least d.*

Proof. Let d and G be as in the statement. We show the statement for white nodes; the proof for black nodes follows by symmetry.

Denote by $n^W_{\geqslant d}$ the number of white nodes with degree at least d, and by $n^W_{<d}$ the number of the rest of white nodes. By the handshake lemma for bipartite graphs, we have

$$|E| \geqslant d n^W_{\geqslant d} + n^W_{<d} = (d-1) n^W_{\geqslant d} + |V_W|. \tag{II.1}$$

As G is a tree, we have $|E| = |V_W| + |V_B| - 1$. Combining this with Equation (II.1) gives the result

$$n^W_{\geqslant d} \leqslant (|V_B| - 1)/(d-1). \qquad \square$$

We can now prove the following lemma showing that each round of DEG reduces the number of nodes by a constant factor:

Lemma II.3.4. *Let $s, t \geqslant 1$, and let $d \geqslant 1$ and let $G = (V_W \sqcup V_B, E)$ be a tree with bipartition (V_W, V_B). Running two iterations of $\mathrm{DEG}(s,t)$ on G reduces the number of nodes in G by a factor of $\Omega(st)$.*

Proof. Let s, t and G be as in the statement. We call nodes of V_W white nodes and nodes of V_B black nodes. Let $L_1 = \mathrm{degen}_{s,t}(G)$, $G_1 = G \setminus L_1$, $L_2 = \mathrm{degen}_{s,t}(G_1)$, and $G_2 = G_1 \setminus L_2$. Let n^W_1 and n^B_1 be the number of white and black nodes in L_1, respectively, and let n^W_2 and n^B_2 be the number of white and black nodes in L_2.

By Lemma II.3.3, we have

$$n^W_1 \leqslant (|V_B| - 1)/(t-1) \text{ and } n^B_1 \leqslant (|V_W| - 1)/(s-1).$$

Applying Lemma II.3.3 again on G_1, we have

$$n^W_2 \leqslant (n^B_1 - 1)/(t-1) \text{ and } n^B_2 \leqslant (n^W_1 - 1)/(s-1).$$

Expanding the definitions of n^W_1 and n^B_1 gives the result:

$$n^W_2 \leqslant |V_W|/\Omega(st) \text{ and } n^B_2 \leqslant |V_B|/\Omega(st). \qquad \square$$

Combining this lemma with the simple fact that all nodes can locally compute $\mathrm{degen}_{s,t}$ for each round, we immediately get the locality of $\mathrm{DEG}(s,t)$:

Corollary II.3.5. *The running time of procedure* DEG(s,t) *is* $O(\log_{st} n) = O(\log_{\max\{s,t\}} n)$ *in the LOCAL model.*

We can now show that resilient problems can be solved efficiently in the LOCAL model:

Lemma II.3.6. *Any* (s,t)*-resilient problem* $\Pi(d,\delta)$ *can be solved with locality* $O(\log_{\max\{s,t\}} n)$.

Proof. Let G be the input tree. Start by computing layer decomposition $L_1, \ldots, L_k$ for G; this can be done with locality $O(\log_{\max\{s,t\}} n)$ by Corollary II.3.5. Label each edge that is between two nodes in the same layer with 0.

We can now proceed to label the rest of the edges layer-by-layer, from $k-1$ to 1. The nodes on layer i label all their adjacent edges in a valid manner. This is possible by Lemma II.3.2 as white (respectively black) nodes in layer i have at most s (respectively t) edges to layer i or higher layers, and only those edges have a label. Note that this procedure can be done in parallel by all nodes of layer i as all edges with both endpoints in the same layer have their output fixed at 0. Finally, by definition of L_1, white nodes on the layer 1 have at most $s < d$ neighbors, and black nodes on the layer 1 have at most $t < \delta$ neighbors, hence they do not impose any restrictions on their adjacent edges. □

Now we are finally ready to prove Lemma II.3.1:

Proof of Lemma II.3.1. Let us focus on the case where $W(d)$ is center-good; the case where $B(\delta)$ is center-good is symmetric. By definition, any sufficiently large set $w \in W(d)$ contains an element $p \in [d^{\varepsilon}, d^{1-\varepsilon}]$. Therefore, the problem is $(d^{\min\{\varepsilon,1-\varepsilon\}}, 1)$-resilient, and the rest follows by Lemma II.3.6. □

II.3.2 Aggressive rake-and-compress

In this section, we introduce the aggressive rake-and-compress algorithm, compute its complexity, and give some of its key properties.

For a graph G and some positive integer r, we define the following two sets:

$$\begin{aligned}
\text{leaves}(G) &= \{v \in V(G) \mid \deg_G(v) = 1\}, \\
\text{ext}_{r,\Delta}(G) &= \{v \in V(G) \mid \forall u \in N^r[v], \deg_G(u) < \Delta\} \\
&= V(G) \setminus N^r[\{v \in V(G) \mid \deg_G(v) \geqslant \Delta\}],
\end{aligned}$$

where $N^r[v]$ in the distance-r closed neighborhood of v. Removing the first set from the graph is a *rake* operation, removing the second set from the graph is a *compress* operation.

By computing these sets recursively and removing them from the graph, we can partition the graph into layers. This is a very helpful tool to create efficient $O(\log_\Delta n)$ local algorithms.

Procedure $\mathrm{ARC}(r, \Delta)$ partitions the set of nodes V into non-empty sets $L_1, L_2, \ldots, L_k$ for some L as follows:

$$\begin{aligned} G_0 &= G, \\ L_{i+1} &= \mathrm{leaves}(G_i) \cup \mathrm{ext}_{r,\Delta}(G_i) \quad \text{if } G_i \text{ is not empty}, \\ G_{i+1} &= G_i \setminus L_{i+1}. \end{aligned}$$

In the following, we show that computing $\mathrm{ARC}(r, \Delta)$ requires only $O(r^2 \cdot \log_{\Delta/r} n)$ rounds.

Lemma II.3.7. *Let $G = (V, E)$ be a tree and $r \in \{1, 2, \ldots\}$. Apply successively the distance-r rake operation, compress operation and then rake operation again, r times successively. If V' is the remaining vertex set then $|V'| \leqslant O(r)|V|/\Delta$.*

Proof. Let G be a tree. We color the nodes of the tree as follows:

- a node of degree Δ is colored blue,
- a non-leaf node of degree $< \Delta$ at distance $\leqslant r$ from a node of degree Δ is colored red,
- and all other nodes are colored white.

Notice that after a rake operation and a compress operation, all white nodes are removed. To account for the remaining rake operation, we recolor all red nodes as follows; see Section II.3.2:

- an old red node is recolored red if it is adjacent to at least two non-white nodes,
- and all the other old red nodes are recolored white.

We do this recoloring r times.

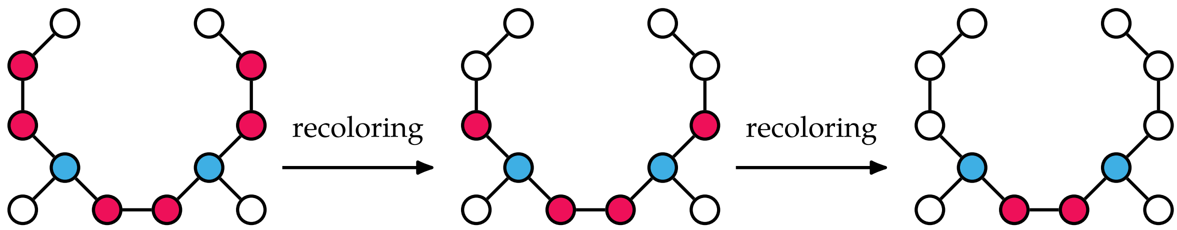


Figure II.2: An example of the initial coloring and two recoloring steps, with $r = 2$ and $\Delta = 3$.

Notice that after the first rake operation and the compress operation, and then the following r successive rake operations, all white nodes are taken. Now, we want to bound the number of leftover blue and red nodes. Let B and R be the sets of blue and red nodes. Now let G' be the subgraph of G induced by blue and red nodes, rooted at an arbitrary blue node. Observe first that every leaf node in G' is a blue vertex. Indeed, suppose for sake of contradiction that there is a red leaf node v. It is red because it was at distance $\leqslant r$ from a blue node b at the beginning. Consider the subtree of G' starting from b. Because its diameter is at most r, this subtree is white at the end of the recoloring.

This is a contradiction with the fact that v is red. Now, we claim the following:

$$|V(G)| \geqslant \Delta|B| + 2, \tag{II.2}$$
$$|R| \leqslant 2r|B|. \tag{II.3}$$

Once those claims are proven, we can combine them and end the proof with the following: $|V'| = |R| + |B| = O(r)|V(G)|/\Delta$. Equation (II.2) follows from the handshake lemma and the fact that G has $|V(G)| - 1$ edges. To prove (II.3), we first need to define the *depth* of a red node of G' as follows: $\operatorname{depth}(v)$ is the minimum distance of v to a blue node by only taking paths of G' that go downward in the tree, i.e. the node of the path closest to the root should be v. By a previous claim, $\operatorname{depth}(v)$ is indeed well-defined for any $v \in R$. Said in other words, it would be the minimum distance from v to a blue node if we orient the edges of G' away from the root.

For any red node v, we claim that $\operatorname{depth}(v) \leqslant 2r$. Indeed, assume for sake of contradiction that there is some node v such that all red downward paths in G' of length $2r+1$ starting from v contain no blue node; see Section II.3.2. Fix one of these paths P and look at the center vertex u of this path. This vertex u splits P, we can write $P = P_1 u P_2$, where $v \in P_1$. We claim that u is at distance at least $r+1$ from a blue node, which is in contradiction with the fact that u is red. To show the claim, assume for sake of contradiction that there exists some downward path $P_3 \subseteq V(G')$ of length $r+1$ from u to a blue node b. Let $P' = P_1 \triangle P_3 \cup \{u\}$ be the path formed by joining P_3 to P_1, and removing all vertices that appear twice (so that it is indeed a path). This is a path of length $\leqslant |P_1| + |P_3| \leqslant r + (r+1) = 2r+1$ from v to a b, contradiction.

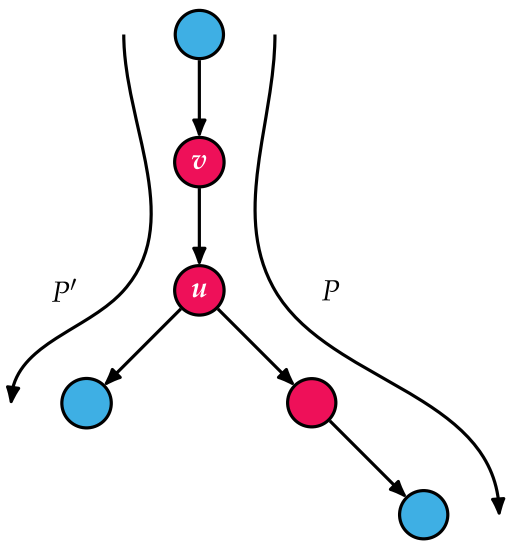


Figure II.3: Explanation of the proof that $\operatorname{depth}(v) \leqslant 2r$ for $r = 1$.

Now, let us count the number of red nodes. Each red node receives one token and sends it to one of its nearest blue successor in G' (break ties arbitrarily). Because of the depth property, each blue node b can receive only $2r$ tokens at most, from every red node in the red downward path that ends at b's parent. Hence, $|R| \leqslant 2d|B|$, concluding the proof. $\square$

Corollary II.3.8. ARC(r, Δ) *can be computed in* $O(r^2 \cdot \log_{\Delta/r} n)$ *rounds.*

Proof. When we apply ARC $r+1$ times, we remove more vertices than by applying the distance-r rake operation, compress operation and then rake operation again, r times successively. Therefore, by Lemma II.3.7, computing $r+1$ ARC layers divides the number of vertices not assigned to a layer by $\Omega(\Delta/r)$. Therefore, after computing $(r+1)k$ layers, the number of vertices not assigned to any layer is divided by $\Omega((\Delta/r)^k)$. We need to compute at most $O(r \cdot \log_{\Delta/r} n)$ layers to have all vertices assigned to some layer. In total, $O(r^2 \cdot \log_{\Delta/r} n)$ LOCAL rounds are needed, because every layer takes $O(r)$ LOCAL rounds to compute. □

Here is a key property that is used to build algorithms optimal in n, d and δ:

Lemma II.3.9. *After the* ARC$(r, d-k)$ *procedure, every relevant degree-d vertex in G is adjacent to k raked vertices (i.e. neighbors which were taken due to their degree being* 1*). Moreover, we can assume those raked vertices are the k lowest-layer neighbors of v.*

Proof. We apply the procedure ARC$(r, d-k)$ to the input G. Denote by $L_1, \ldots, L_m$ the resulting layer partition. Let v be a degree-d node on layer i. Consider the set R of the k lowest-layer neighbors of v. When $u \in R$ is added to its layer, it cannot be due to compress operation, because when at this time, v had at least $d-k$ other neighbors still not assigned to a layer yet, i.e. v had residual degree $\geqslant d-k+1$. Therefore, one of its neighbors cannot be added a layer at that time, and we get a contradiction. Hence, u gets assigned to a layer because of a rake operation. □

For $r \geqslant 2$, we have the more general statement:

Lemma II.3.10. *Given a decomposition given by the* ARC$(r, d-k)$ *procedure with $r \geqslant 2$, we have the following. For every relevant degree-d vertex v, and every vertex u in the set of k lowest-layer neighbors of v, every neighbor of u different from v is a raked vertex. Moreover, u is also raked.*

Proof. We apply the procedure ARC$(r, d-k)$ to the input G, with $r \geqslant 2$. Denote by $L_1, \ldots, L_m$ the resulting layer partition. Let v be a relevant degree-d vertex, and let u be a layer-j vertex in the set of k lowest-layer neighbors of v. Let w be a neighbor of u different from v, in layer i. u must be a raked vertex, because it is one of the k lowest-layer neighbors of v. Therefore, w is at a lower layer than u, i.e. $i < j$. We also get that v is of degree $\geqslant d-k+1$ in G_j, as u, one of the k lowest-layer neighbors of w, is still in G_j. Therefore, w is at distance 2 of v in G_i, so w must be a raked vertex because v is of degree $\geqslant d-k+1$ in G_i. □

We will now use these tools to prove upper-bound results for two problems that play a key role in our classification.

II.3.3 Bipartite factor problem

Bipartite (k,l)-factor is the problem $\Pi(d,\delta) = (d,\delta,\{k\},\{l\})$.

Note that bipartite $(1,1)$-factor is the same problem as bipartite perfect matching.

Lemma II.3.11. *When k and l are non-zero constants, bipartite (k,l)-factor can be solved in $O(\log_\delta n)$ rounds.*

Proof. We apply the procedure $\mathrm{ARC}(1,\min\{d-k,\delta-l\})$ on the input graph G. We therefore have access to Lemma II.3.9. Let $L_1,\ldots,L_m$ denote the resulting layer partition. The general idea is to iterate through the layers in a backwards order, and match a relevant node with some of its raked neighbors (exist by Lemma II.3.9). More formally, consider the nodes on layer i, and assume that all edges incident to nodes of layers greater than i have labeled. Vertices in L_i perform the following steps to label all their incident edges that remain unlabeled:

- Any edge between two nodes in L_i is labeled with 0,
- Any white (resp. black) node that already has one incident edge labeled 1 chooses its $k-1$ (resp. $l-1$) lowest-layer neighbors, labels the unlabeled edges to these neighbors by 1, and labels the rest of the unlabeled incident edges by 0. If those neighbors do not all exist, then v is not relevant by Lemma II.3.9, and we can label all unlabeled incident edges of v by 0.
- Any white (resp. black) node that already has one incident edge labeled 1 chooses its k (resp. l) lowest-layer neighbors, labels the unlabeled edges to these neighbors by 1, and labels the rest of the unlabeled incident edges by 0. If those neighbors do not all exist, then v is not relevant by Lemma II.3.9, and we can label all unlabeled incident edges of v by 0.

Let M be the set of edges labeled with 1. We have to argue that this case distinction cover all possible cases, that is, every relevant vertex v has at most one higher-layer neighbor u such that $uv \in M$. This is true, because if v is a relevant vertex with a higher-layer neighbor u such that $uv \in M$, then by definition of the algorithm v is a raked vertex and cannot have more than one edge to a higher-layer vertex. Notice that we can ignore vertices in L_1: they are not relevant because they are not of degree d nor δ. Furthermore, by definition, every relevant white (resp. black) node v is incident k (resp. l) edges in M. Therefore, we proved the correctness of the algorithm. Additionally, the desired round complexity is achieved, as we run $\mathrm{ARC}(1,\min\{d-k,\delta-l\})$ in $O(\log_\delta n)$ time by Corollary II.3.8 (and because k,l are constants) and go through all the $O(\log_\delta n)$ layers once in time $O(\log_\delta n)$. □

II.3.4 Quasi-orientation problem

Quasi-(k,l)-orientation is the problem $\Pi(d,\delta) = (d,\delta,\{k\},\{0,\delta-l\})$.

This problem is easiest to understand if we interpret it so that the edges in X are oriented towards black nodes and edges in $E \setminus X$ are oriented towards white nodes. Then in a quasi-(k,l)-orientation, all relevant white nodes have outdegree k while all relevant black nodes have outdegree l or δ.

Lemma II.3.12. *When k is a constant, quasi-(k,l)-orientation can be solved in $O(\log_d n)$ rounds.*

Proof. We apply the procedure $\mathrm{ARC}(2, d-k)$ to the input graph G. Let $L_1, \ldots, L_m$ denote the resulting layer partition. The general idea is to iterate through the layers in a backwards order, and orient a white degree-d node towards some of its black raked neighbors, and orient a black degree-δ node towards some of its white raked neighbors (only if the black node is adjacent to an edge labelled with a 1). More formally, consider the nodes on layer i, and assume that all edges incident to nodes of layers $> i$ have labeled. Vertices in L_i perform the following steps to label all their incident edges that remain unlabeled:

- If a black node is not incident to any edge labeled 1, label all unlabeled incident edges by 0.
- If a black node is incident to exactly one edge labeled 1, choose its l lowest-layer neighbors, label the unlabeled edges to these neighbors by 0, and label the rest of the unlabeled incident edges by 1. These lower-layer neighbors always exist because the black node is a raked vertex.
- If a white node is incident (respectively not incident) to an edge labeled 1, choose its $k-1$ (respectively k) lowest-layer neighbors, label the unlabeled edges to these neighbors by 1, and label the rest of the unlabeled incident edges by 0. In the case where those neighbors do not all exist, then v is not relevant by Lemma II.3.9, and we can label all unlabeled incident edges of v by 0.

Let M be the set of edges labeled with 1. We have to argue that this case distinction cover all possible cases, that is, every relevant vertex v has at most one higher-layer neighbor u such that $uv \in M$. For relevant black vertices, this is because black vertices with an edge labeled 1 that goes to a higher-layer neighbor are raked vertices. Now, let us tackle the case of white vertices. Suppose a white vertex $u \in L_i$ is adjacent to a higher-layer or equal-layer black vertex $v \in L_j$, with all but l of v's incident edges labeled 1. By Lemma II.3.10, v must be a raked vertex, and one the k lowest-layer neighbors of some relevant node $w \in V(G_j)$. We also get that u is a raked vertex. Thus, u cannot have 2 incident edges labeled with 1 coming from higher-layer neighbors. Notice that we can ignore white vertices in L_1: they are not relevant because they are of degree $< d$. Black vertices in L_1 have no relevant white neighbor, so we can satisfy them trivially. Other relevant black nodes either have $\delta - l$ incident edges labeled 1 or all their incident edges labeled 0, and furthermore, by definition, every white relevant node (thus not in L_1) is incident to k edges in M. Therefore, the algorithm is correct. Additionally, the desired round complexity is achieved, as we run $\mathrm{ARC}(2, d-k)$ in $O(\log_d n)$ time by Corollary II.3.8 (and because k and l are constant) and go through all the $O(\log_d n)$ layers once in time $O(\log_d n)$. □

II.4 Classification of logarithmic problems

We now have all the tools to classify logarithmic problems and prove our main theorem. A summary of the classification is displayed in Section II.4.

Theorem II.1.1. *Let $\Pi(d, \delta)$ be a structurally simple family of binary labeling problems and assume that the complexity of $\Pi(d, \delta)$ in trees is $\Theta_{d,\delta}(\log n)$. Then the complexity of $\Pi(d, \delta)$ in trees for each d and δ falls in one of these classes, as long as $\delta \leqslant d = O(n^{1-\alpha})$ for some $\alpha > 0$:*

1. $\Theta(\log n)$,
2. $\Theta(\log_\delta n)$,
3. $\Theta(\log_d n)$.

Proof. Fix $\delta \leqslant d$, let $w = \hat{W}(d)$ and $b = \hat{B}(\delta)$. Let Π solvable with $\Theta_{d,\delta}(\log n)$ complexity. We know by Lemma II.2.1 that the problem has complexity $\Omega(\log_d n)$. However, this is not tight: there are some problems that cannot be solved in $O(\log_d n)$ time. We make a case distinction to classify the problem complexity into multiple classes. First, if w is center-good, then it can be solved in $O(\log_d n)$ rounds using Lemma II.3.6. Now suppose for the rest of the proof that w is edge-good. Let $w = s_w w_1 0^k w_2 t_w$ with $|w_1|, |w_2| \leqslant C$, and $b = s_b \dots t_b$. The complexities split into different classes depending on w and b.

- If $s_w = s_b = 1$ or $t_w = t_b = 1$, then problem is of constant complexity.
- Else if $s_b = t_b = 1$, then $\Pi(d, \delta)$ can be solved with the quasi-$(k, 0)$-orientation algorithm of Lemma II.3.12 (and possibly Lemma II.2.3) in time $O(\log_d n)$, because it cannot be that w_1 and w_2 only contains 0's, otherwise we would have $w = 0^{d+1}$ (as $s_w = t_w = 0$).
- Else, if none of the above apply, either $s_b = 0$ or $t_b = 0$. We will handle this case for the rest of the proof.

Without loss of generality, assume in the following that $t_b = 0$ (otherwise, one can use Lemma II.2.3 and get back to the previous case). If $s_b = 1$, it cannot be that $b = 10^*$, as $s_w = 0$, as the problem would be unsolvable. Therefore, b contains at least 2 1's and $\Pi(d, \delta)$ can be solved in $O(\log_d n)$ rounds with the quasi-(k, l)-orientation algorithm.

We are now left with the case $s_b = t_b = 0$ for the rest of the proof. Then the problem has complexity $\Omega(\log_\delta n)$ by Proposition II.2.7 (assuming $d = O(n^{1-\alpha})$ for some $\alpha > 0$). Moreover, by assumption, the complexity is $O(\log n)$. There are still two possible complexity classes: $\log_\delta n$ and $\log n$. To finish our classification, we need to prove that in all cases, the complexity is either $\Omega(\log n)$ or $O(\log_\delta n)$. If b is center-good, then the complexity is $O(\log_\delta n)$ by Lemma II.3.6. For the rest of the proof, suppose b is edge-good. Let $b = s_b b_1 0^l b_2 t_b$ with $|b_1|, |b_2| \leqslant C$.

- If $s_w = t_w = 1$, then $\Pi(d, \delta)$ can be solved in time $O(\log_\delta n)$ with Lemma II.2.2, possibly Lemma II.2.3, and the quasi-$(k, 0)$-orientation algorithm of Lemma II.3.12. The algorithm is applicable because b_1 and b_2 do not only contains 0's, otherwise we would have $b = 0^{\delta+1}$.
- Else, either $s_w = 0$ or $t_w = 0$.

For the rest of the proof, we have $s_w = 0$ or $t_w = 0$, and $s_b = t_b = 0$. There are multiple cases:

- Type A: there exists $i \in \{1, 2\}$ such that both b_i and w_i contain a 1. The complexity of a type A problem is $O(\log_\delta n)$ by Lemma II.3.11 and using Lemma II.2.3 (if necessary).
- Type B: w_1 and b_2 only contain 0's, and w_2 and b_1 contains a 1. If $s_w = 0$, the problem complexity is $\Omega(\log n)$ by Proposition II.2.5. Else, $s_w = 1$ and $t_w = 0$, and the complexity is $O(\log_\delta n)$ using Lemma II.3.12 and Lemma II.2.2.
- Type C: w_2 and b_1 only contain 0's, and w_1 and b_2 contains a 1. Using Lemma II.2.3, we can get from type C back to type B without loss of generality.
- Type D_c: there exists $c \in \{w, b\}$ such that both c_1 and c_2 only contain 0's. A type D_b or D_w problem is unsolvable. Consider a type D_b problem. Then the problem is unsolvable because $b = 0^{\delta+1}$. Now consider a type D_w problem. If $s_w = t_w = 0$, the problem is unsolvable because $w = 0^{d+1}$. Else, one of s_w and t_w is equal to 0, but not both. So $w = 0^+01$ or $w = 100^+$ and as $s_b = t_b = 0$, the problem is unsolvable. $\square$

II.5 Outside logarithmic region

In Sections II.2 to II.4 we have analyzed problems of complexity $\Theta_{d,\delta}(\log n)$. In this section we explore the two remaining cases of solvable problems: classes $O_{d,\delta}(1)$ and $\Theta_{d,\delta}(n)$. All results are summarized in Section II.1.7.

II.5.1 Constant complexity

Proposition II.5.1. *If the complexity of $\Pi(d, \delta)$ is $O_{d,\delta}(1)$, then it is also* 1.

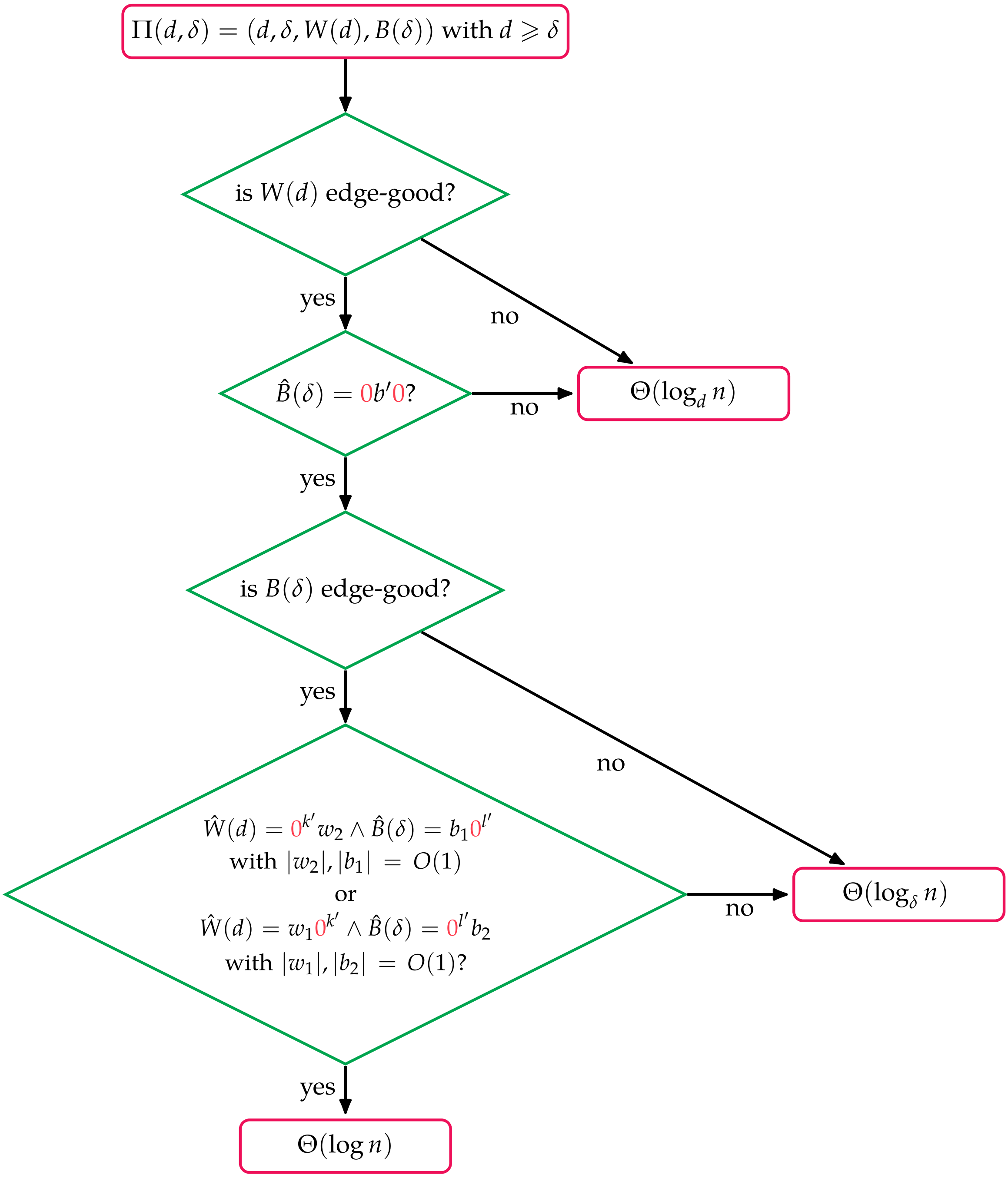


Figure II.4: Complexity flowchart for logarithmic problems; we assume here that the complexity of $\Pi(d, \delta)$ is $\Theta_{d,\delta}(\log n)$.

Proof. If the complexity of the problem is $O_{d,\delta}(1)$, then by the classification of [BBE+20a] the problem is of the form $(111^+, B)$ with B non-empty or of the form (W, B) with $0 \in W \cap B$ (or their switch or reverse). To solve the first type of problem in 1 round, one can run an algorithm where black nodes label k neighboring edges with a 1 (if $k \in B$), and the rest of the neighboring edges with a 0. To solve the second type of problems in 1 round, one can run an algorithm where every node labels every neighboring edge with a 0. Both of these algorithms are correct and run in 1 round. □

II.5.2 Linear complexity: upper bounds

For a bipartite graph G with bipartition $V(G) = V_W \sqcup V_B$, denote $W_s = \{v \in V_W \mid \deg(v) = s\}$ and $B_t = \{v \in V_B \mid \deg(v) = t\}$.

Lemma II.5.2. *Let G be a tree with bi-partition $V(G) = V_W \sqcup V_B$. Then $G[W_s \cup B_t]$ has components of diameter at most $4|V(G)|/(s+t)+1$.*

Proof. Let G' be a connected component of $G[W_s \cup B_t]$. Let d be the diameter of G' and let P be a path of length d in G'. Notice that in G', vertices are either in W_s or in B_t. Therefore, P contains at least $(d-1)/2$ vertices from W_s and at least $(d-1)/2$ vertices from B_t. By the handshake lemma, $2|V(G)| \geqslant s|W_s| + t|B_t| \geqslant (s+t)(d-1)/2$. Finally, we get $d \leqslant 4|V(G)|/(s+t)+1$. □

Corollary II.5.3. *Any solvable binary labeling problem $\Pi(d,\delta)$ has complexity $O(n/(d+\delta))$.*

Proof. Let (G, d, δ) be an instance of Π with $W \sqcup B$ a bipartition of $V(G)$. As we only have constraints on vertices of W of degree d and on vertices of B of degree δ, one only needs to solve Π on the connected components of $G[W_d \cup B_\delta]$. By Lemma II.5.2, the diameter is at most $4|V(G)|/(d+\delta)+4$, and every LOCAL problem can be solved in a number of rounds equal to the diameter. Therefore, Π can be solved in time $O(n/(d+\delta))$. □

II.5.3 Linear complexity: lower bounds

Up to switching, there are two types of problems with complexity $\Omega_{d,\delta}(n)$ [BBE+20a]:

- Type A: $(W, B) = (10^+1, 010)$,
- Type B: $(W, B) = (*10^+, 0^+1*)$,

where $*$ is a placeholder for an arbitrary bit. We prove that both types have complexity $\Omega(n/(d+\delta))$. Notice that in the first type, $\delta = 3$ so the complexity is $\Omega(n/d)$.

Lemma II.5.4. *If $\Pi(d,3)$ is a problem of type A, then $\Pi(d,3)$ has complexity $\Omega(n/d)$.*

Proof. This follows from the proof of Theorem 8.1 of [BBE+20a]. In this proof, the problem is reduced to 2-coloring of a path of size n/d, and it is known that the complexity of 2-coloring a path of length ℓ is $\Theta(\ell)$. Hence, $\Pi(d,3)$ has complexity $\Omega(n/d)$ □

Lemma II.5.5. *If $\Pi(d,\delta)$ is of type B, then $\Pi(d,\delta)$ has complexity $\Omega(n/(d+\delta))$.*

Proof. This follows the proof of Theorem 8.2 of [BBE+20a]. In this proof, the problem is reduced to a problem called *almost oriented path* for a path of size $f(n,d,\delta)$, and then it is showed that the complexity of almost orienting a path of length ℓ is $\Theta(\ell)$. The almost oriented path problem takes as input a properly 2-colored path and asks to orient the edges such that at most one node of degree 2 has all its incident edges outgoing, and all other nodes of degree 2 have exactly one outgoing edge (nodes of degree 1 are unconstrained).

Let us compute $f(n,d,\delta)$. In the proof, of Theorem 8.2, the authors start from properly 2-colored path, connect to each white node $d-2$ new black leaves, and to each black node $\delta-2$ new white leaves. Then, the authors connect an additional node to each endpoint of the path while keeping a proper 2-coloring. Then, any solution for $\Pi(d,\delta)$ for this new instance can be mapped to a solution for the almost oriented path problem on the original instance. Let k be the order of the original instance. n is the size of the new instance so $n \geqslant (k/2+1)(\delta-2) + (k/2+1)(\delta-2)$ and therefore $k = \Omega(n/(d+\delta))$. This finishes the proof that $\Pi(d,\delta)$ has complexity $f(n,d,\delta) = \Omega(k) = \Omega(n/(d+\delta))$. □

II.5.4 Linear complexity: summary

By putting together Sections II.5.2 and II.5.3, we obtain:

Corollary II.5.6. *If the complexity of $\Pi(d,\delta)$ is $\Theta_{d,\delta}(n)$, then it is also $\Theta(n/(d+\delta))$.*

II.6 Structurally simple problems and context-free languages

We have now completed the classification of structurally simple binary labeling problems. In this section, we show that all binary labeling problems that can be defined with context-free grammar are structurally simple:

Lemma II.1.4. *Let $X(k) \subseteq \{0,1,\dots,k\}$ be a family of sets. If $\hat{X}$ is a context-free language, then $X(k)$ is structurally simple.*

To prove this, we need the following result from language theory:

Theorem II.6.1 ([Ili94, Theorem 2.1]). *Let B a constant and L a context-free language such that for every length n of a word in L, there are at most B words of length n in L. Then L can be written as a finite union of so-called paired loops, i.e.*

$$L = \bigcup_{i \in I} \{u_i v_i^n w_i x_i^n y_i \mid n \geqslant 0\}$$

for some finite set I and words u_i, v_i, w_i, x_i, y_i for every $i \in I$.

We also need the following auxiliary lemma:

Lemma II.6.2. *For every language $L = \{uv^n wx^n y \mid n \geqslant 0\}$, for some words $u, v, w, x, y \in \{0, 1\}^*$, there exist a real $\alpha > 0$ and integers B, C, N such that for $n \geqslant N$, at least one of these is true:*

1. *L contains a word z with length n with a 1 at position i s.t. $\alpha n - B \leqslant i \leqslant (1-\alpha)n + B$,*
2. *there is no word of L of length n with a 1 at position i s.t. $i > C$ or $i < n - C$.*

Proof. Let $N = |uvwxy|$. Let $C = \max\{|u|, |v|, |w|, |x|, |y|\}$. If $v, w, x \in 0^*$, all the words $z \in L$ of length $n \geqslant N$ can only contain symbol 1's at positions $i \in [0, C] \cup [|z| - C, |z|]$. Otherwise, we know that the word vwx contain a symbol 1. Let $n \geqslant N$. Take $z \in L$ of length n and let k be an integer such that $z = uv^k wx^k y = uv^{k-1}(vwx)x^{k-1}y$. v^{k-1} and x^{k-1} exist, because $n \geqslant N$. Therefore, there is a 1 in z, inside the middle vwx, at position

$$i \in [|u| + (k-1)|v|, |z| - |y| - (k-1)|x|].$$

There are multiple cases. We first handle the case where v and x are not empty. If this is the case, then let $\alpha = \min\{|v|, |x|\}/(|v| + |x|)$ and $B = \alpha|uvwxy|$. Then $\alpha n - B = \alpha((k-1)|v| + (k-1)|x|) = (k-1)\alpha(|v| + |x|) \leqslant (k-1)|v|, (k-1)|x|$. This means that $i, |z| - i \geqslant \alpha n - B$. Let us now handle the case where v is empty; the proof when x is empty is similar, and v and x cannot be both empty. Let $C = 2\max\{|u|, |v|, |w|, |x|, |y|\}$, $\alpha = 1/2$, and $B = |uvwxy|$. If $x \in 0^*$, then z only contain symbol 1's at positions $i \in [0, C] \cup [|z| - C, |z|]$. Otherwise, write $z = uwx^{\lfloor k/2 \rfloor} x x^{\lceil k/2 \rceil - 1} y$. Notice that there is a 1 in z, inside the middle x, at position

$$i \in \big[|u| + |w| + \lfloor k/2 \rfloor |x|, |z| - |y| - (\lceil k/2 \rceil - 1)|x|\big].$$

Then $\alpha n - B \leqslant (k-2)|x|/2 \leqslant \lfloor k/2 \rfloor |x|, (\lceil k/2 \rceil - 1)|x|$. This means that $i, |z| - i \geqslant \alpha n - B$. This concludes the proof. □

Proof of Lemma II.1.4. Using Theorem II.6.1, we can write

$$\hat{X} = \bigcup_{i \in I} \{u_i v_i^k w_i x_i^k y_i \mid k \geqslant 0\}$$

for some finite set I and words u_i, v_i, w_i, x_i, y_i for every $i \in I$. Apply Lemma II.6.2 for each $L_i = \{u_i v_i^k w_i x_i^k y_i \mid k \geqslant 0\}$ separately, and obtain constants α_i, B_i, C_i, and N_i for each $i \in I$, so that for every $i \in I$, $n \geqslant N_i$, and $w \in L_i$ of length $n \geqslant N_i$ at least one of these is true:

1. L_i contains a word w with length n with a 1 at position x s.t. $\alpha_i n - B_i \leqslant x \leqslant (1-\alpha_i)n + B_i$,
2. there is no word of L of length n with a 1 at position x s.t. $x > C_i$ or $x < n - C_i$.

Let $\alpha = \min\{\alpha_i \mid i \in I\}$, $B = \max\{B_i \mid i \in I\}$, $C = \max\{C_i \mid i \in I\}$, and $N = \max\{N_i \mid i \in I\}$. Then, given some $n \geqslant N$, at least one of these is true:

1. $\hat{X}$ contains a word $w \in L_i$ for some $i \in I$ with length n with a 1 at position x s.t. $\alpha n - B \leqslant x \leqslant (1-\alpha)n + B$, because $\alpha n - B \leqslant \alpha_i n - B_i$ and $(1-\alpha_i)n + B_i \leqslant (1-\alpha)n + B$,
2. there is no word of $\hat{X}$ of length n with a 1 at position x s.t. $x > C$ or $x < n - C$, because for every $i \in I$, $C \geqslant C_i$ or $n - C \leqslant n - C_i$.

This proves that for any $\varepsilon > 0$, there exists some N such that for any word $w \in \hat{X}$ with length $n \geqslant N$, at least one of these is true:

1. word w is ε-center-good: it has a 1 at position $p \in [n^\varepsilon, n^{1-\varepsilon}]$,
2. word w is C-edge-good: it has 1's only at positions $p \in [0, C] \cup [|w| - C, |w|]$.

This means that X is structurally simple. □

References

[BBE+20a] Alkida Balliu et al. “Classification of Distributed Binary Labeling Problems”. In: *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*. Ed. by Hagit Attiya. Vol. 179. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 17:1–17:17. DOI: 10.4230/LIPIcs.DISC.2020.17.

[BBE+20b] Alkida Balliu et al. *Classification of distributed binary labeling problems*. 2020. DOI: 10.48550/arXiv.1911.13294. arXiv: 1911.13294 [cs.DC]. URL: https://arxiv.org/abs/1911.13294.

[BBOS20] Alkida Balliu et al. “How much does randomness help with locally checkable problems?” In: *PODC ’20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*. Ed. by Yuval Emek and Christian Cachin. ACM, 2020, pp. 299–308. DOI: 10.1145/3382734.3405715.

[BBOS21] Alkida Balliu et al. “Almost global problems in the LOCAL model”. In: *Distributed Computing* 34.4 (2021), pp. 259–281. DOI: 10.1007/S00446-020-00375-2.

[BFH+16] Sebastian Brandt et al. "A lower bound for the distributed Lovász local lemma". In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. Ed. by Daniel Wichs and Yishay Mansour. ACM, 2016, pp. 479–488. DOI: 10.1145/2897518.2897570.

[BHK+18] Alkida Balliu et al. "New classes of distributed time complexity". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. Ed. by Ilias Diakonikolas, David Kempe, and Monika Henzinger. ACM, 2018, pp. 1307–1318. DOI: 10.1145/3188745.3188860.

[BKK+23] Alkida Balliu et al. "Sinkless Orientation Made Simple". In: *2023 Symposium on Simplicity in Algorithms, SOSA 2023, Florence, Italy, January 23-25, 2023*. Ed. by Telikepalli Kavitha and Kurt Mehlhorn. SIAM, 2023, pp. 175–191. DOI: 10.1137/1.9781611977585.ch17.

[CKP19] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. "An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model". In: *SIAM Journal on Computing* 48.1 (2019), pp. 122–143. DOI: 10.1137/17M1117537.

[CP19] Yi-Jun Chang and Seth Pettie. "A Time Hierarchy Theorem for the LOCAL Model". In: *SIAM Journal on Computing* 48.1 (2019), pp. 33–69. DOI: 10.1137/17M1157957.

[FG17] Manuela Fischer and Mohsen Ghaffari. "Sublogarithmic Distributed Algorithms for Lovász Local Lemma, and the Complexity Hierarchy". In: *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*. Ed. by Andréa W. Richa. Vol. 91. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 18:1–18:16. DOI: 10.4230/LIPIcs.DISC.2017.18.

[FLS+95] Zoltán Füredi et al. "Graphs of Prescribed Girth and Bi-Degree". In: *Journal of Combinatorial Theory, Series B* 64.2 (1995), pp. 228–239. DOI: 10.1006/jctb.1995.1033.

[GHK18] Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. "On Derandomizing Local Distributed Algorithms". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by Mikkel Thorup. IEEE Computer Society, 2018, pp. 662–673. DOI: 10.1109/FOCS.2018.00069.

[GS17] Mohsen Ghaffari and Hsin-Hao Su. "Distributed Degree Splitting, Edge Coloring, and Orientations". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel*

Porta Fira, January 16-19. Ed. by Philip N. Klein. SIAM, 2017, pp. 2505–2523. doi: 10.1137/1.9781611974782.166.

[Ili94] Lucian Ilie. "On a Conjecture about Slender Context-Free Languages". In: *Theoretical Computer Science* 132.2 (1994), pp. 427–434. doi: 10.1016/0304-3975(94)00042-5.

[LPS24] Henrik Lievonen, Timothé Picavet, and Jukka Suomela. "Distributed Binary Labeling Problems in High-Degree Graphs". In: *Structural Information and Communication Complexity*. Ed. by Yuval Emek. Cham: Springer Nature Switzerland, 2024, pp. 402–419. isbn: 978-3-031-60603-8.

[MR85] Gary L. Miller and John H. Reif. "Parallel Tree Contraction and Its Application". In: *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*. IEEE Computer Society, 1985, pp. 478–489. doi: 10.1109/SFCS.1985.43.

[NS95] Moni Naor and Larry J. Stockmeyer. "What Can be Computed Locally?" In: *SIAM Journal on Computing* 24.6 (1995), pp. 1259–1277. doi: 10.1137/S0097539793254571.

[PS95] Gheorghe Paun and Arto Salomaa. "Thin and Slender Languages". In: *Discrete Applied Mathematics* 61.3 (1995), pp. 257–270. doi: 10.1016/0166-218X(94)00014-5.

[RG20] Václav Rozhon and Mohsen Ghaffari. "Polylogarithmic-time deterministic network decomposition and distributed derandomization". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*. Ed. by Konstantin Makarychev et al. ACM, 2020, pp. 350–363. doi: 10.1145/3357713.3384298.

[Sch26] Gustav Schmid. "LCLs Beyond Bounded Degrees". In: *arXiv preprint arXiv:2602.02340* (2026).

Part B

Approximating in the LOCAL model: aiming for constants

Chapter III

Distributed algorithms beyond planar graphs: MDS and other problems

Abstract

We prove that given any α-approximation LOCAL algorithm for Minimum Dominating Set (MDS) on planar graphs, we can construct an $f(g)$-round $(3\alpha + 1)$-approximation LOCAL algorithm for MDS on graphs embeddable in a given Euler genus-g surface. Heydt et al. [*European Journal of Combinatorics* (2025)] gave an algorithm with $\alpha = 11 + \varepsilon$, from which we derive a $(34 + \varepsilon)$-approximation algorithm for graphs of genus g, therefore improving upon the current state of the art of $24g + O(1)$ due to Amiri et al. [*ACM Transactions on Algorithms* (2019)]. It also improves the approximation ratio of $91 + \varepsilon$ due to Czygrinow et al. [*Theoretical Computer Science* (2019)] in the particular case of orientable surfaces.

We generalize this result into two directions: (1) by considering other graph problems studied in Distributed Computing such as Minimum k-Tuple Dominating Set, for which constant-round approximation algorithms were known for planar graphs, but not for graphs of bounded genus; and (2) by considering graph classes beyond bounded genus graphs, called *locally nice*, and relying on the asymptotic dimension of the class. We prove these results by a series of meta-theorems about *cuttable* minimization problems with constant-round approximation LOCAL algorithms. Roughly speaking, in cuttable problems, one can systematically extract small subgraphs whose solutions are in proportion to the global solution restricted to the neighborhood of the subgraph.

Acknowledgements

This chapter is a version of a work with Marthe Bonamy, Avinandan Das, Cyril Gavoille, Alexandra Wesolek, and Jukka Suomela, adapted to the thesis. This work has been accepted to the upcoming PODC'26 conference. The authors would like to gratefully acknowledge inspiring discussions with Linda Cook, Carla Groenland and Sergey Norin.

Contents

III.1 Introduction

MINIMUM DOMINATING SET (MDS) is a famous minimization problem on graphs, known to be NP-complete even in cubic planar graphs [GJ79, KYK80]. The goal is to find a smallest subset of vertices of the input graph that intersects all radius-1 balls of the graph.

In this chapter, we focus our attention on distributed algorithms that can approximate MDS on the graph underlying the topology of the network. Due to the covering property of a dominating set, the problem and its variants (like CONNECTED DOMINATING SET [WAF02]) get increasingly more attention in Networking, in particular for mobile and ad-hoc networks. Not only are mobile networks important for point-to-point communications, but specialized ad-hoc networks, such as sensor networks, are important for environmental monitoring tasks. We refer to [KW05, RWZ09] for extended discussions about the motivations of MDS for routing purposes of such networks.

In the LOCAL model, approximating MDS up to a constant factor in general n-vertex graphs is known to require $\Omega(\sqrt{\log n / \log\log n}\,)$ rounds [KMW16, CL21]. On the positive side, it is possible to $(1+\varepsilon)$-approximate MDS for any graph in $\mathrm{poly}(\varepsilon^{-1}\log n)$ rounds by combining the techniques of [GKM17] and of [RG20, Corollary 3.11].

For specific graphs, better round complexities can be achieved. We refer to [LPW13, Suo13] for a large collection of results (including unit-disk graphs, and planar graphs as a basis of the Gabriel graph model widely used in ad-hoc networks [WY07]). For instance, $O(\log^* n)$ rounds suffice for planar graphs [CHW08], and also more generally for K_t-minor-free graphs [CHW18] and for sub-logarithmic expansion graphs[1] [ASS19], and $o(\log^* n)$ rounds are not sufficient to get a $(1+\varepsilon)$-approximation of MDS on cycles [CHW08] or unit-disk graphs [LW08]. So, achieving constant-round algorithms must be at the price of relaxing the $(1+\varepsilon)$ approximation ratio.

For planar graphs, the quest for constant-approximation and constant-round algorithms seems to start with [LOW08], followed by [Len11, Chp. 13] and [LPW13], with a 130-approximation. Since then, a long line of research has been aimed at improving this ratio. In short, the best approximation ratio to date is $11+\varepsilon$, due to [HKO+25]. We refer to the nice survey of [HKO+25] for planar graphs and subfamilies, including lower bounds.

In the meantime, the quest for constant-approximation ratio and constant-round distributed algorithms has been proposed for larger classes of graphs. However, very few examples are known of graph classes $\mathscr{C}(p)$ depending on some fixed parameter p where MINIMUM DOMINATING SET can be solved by a LOCAL algorithm with round complexity $f(p)$ and truly-constant approximation ratio (independent of p).

Larger graph classes that make good candidates for extending pla-

[1] Roughly speaking, the expansion of a graph G is a function f that, for any r, bounds the maximum edge density of a depth-r minor of G by $f(r)$, where a depth-r minor is a minor of G where each branch set has radius at most r.

nar graphs are H-minor-free graphs, with some graph H not limited to K_5 or $K_{3,3}$. For K_p-minor-free graphs, we only know of an exponential approximation ratio in p [KSV21, HKO+25], whereas a linear approximation ratio could be possible. For $K_{3,p}$-minor-free graphs [HKO+25], the dependency is much better (linear in p), but still not truly constant. Very recently, it was shown [BGPW25] (see also Chapter IV) that, for $K_{2,p}$-minor-free graphs, the approximation ratio is 50 regardless of the value of p.

On the other end of the spectrum, the narrowest way to meaningfully extend planar graphs is through embeddable graphs[2] on surfaces[3] of Euler genus g. Such surfaces can be obtained from a sphere by adding $g/2 \geqslant 0$ handles[4] (if they are orientable) or by adding $g \geqslant 1$ cross-caps[5] (if they are non-orientable). The *Euler genus* of a graph G is the minimum number g such that G embeds on a surface of Euler genus g. In this context, [ASS19, Theorem 3.11] proposed a $(24g + O(1))$-approximation algorithm with constant-round complexity (actually linear in g). For graphs of orientable genus g, i.e., embeddable on an orientable surface of genus g, [CHWW19] designed a constant-round algorithm that returns a dominating set of size at most $91\mathrm{M} + 76g - 66$, where M is the optimal size. By adding a simple extra brute-force step in their algorithm, we can convert it into a $(91 + \varepsilon)$-approximation (cf. Proposition III.8.2), so with a ratio independent of g. However, the result of [CHWW19] does not transfer to graphs of bounded Euler genus, as for any g, there are graphs embeddable on the projective plane[6] that cannot be embedded on any orientable surface of genus g (so of orientable genus at least $g + 1$). So, for these graphs, the approximation ratio will be $91 + \varepsilon$ but the number of rounds possibly unbounded (possibly linear in the number of vertices).

[2] I.e., that can be drawn on the surface without edge crossing, like planar graphs for the sphere.

[3] That are compact, connected 2-dimensional manifold without boundary.

[4] By adding a handle, we mean that two disjoint disks of the sphere are replaced by a cylinder.

[5] By adding a cross-cap, we mean that a disk of the sphere is replaced by a Möbius strip.

[6] For instance, a cycle with $n = 2g + 6$ vertices where opposite vertices are connected by an extra edge, cf. [ABY63].

Graphs of Euler genus-g are included in $K_{3,2g+3}$-minor-free graphs, because the Euler genus of $K_{3,2g+3}$ is at least $g + 1$ (cf. [MT01, Theorem 4.4.7]) and the class of Euler genus-g graphs is closed under taking minors. We observe that the approximation ratio for $K_{3,p}$-minor-free graphs, for $p = 2g + 3$, due to [HKO+25, Theorem 2.2] provides an approximation ratio of $O(\sqrt{g})$. Indeed, the ratio of their algorithm is precisely $(2 + \varepsilon)(2\nabla_1 + 1)$, where ∇_1 is the maximal edge density of a depth-1 minor of any graph of the class. As we can check (cf. Proposition III.8.3) that $\nabla_1 \leqslant \sqrt{3g/2} + 3$, the approximation ratio is at most $4\sqrt{3g/2} + 14 + \varepsilon$. As it may occur that $\nabla_1 \geqslant \sqrt{3g/2} - O(1)$ for some Euler genus-g graph (cf. Proposition III.8.3), the best known approximation ratio for MDS in Euler genus-g graphs is $\Theta(\sqrt{g})$, and $91 + \varepsilon$ for orientable genus-g graphs.

It is important to note that no LOCAL algorithms for Euler genus-g graphs can achieve simultaneously constant round complexity and constant approximation ratio. Indeed, any constant round approximation must have ratio $\Omega(\sqrt{\log g / \log\log g})$ as the $\Omega(\sqrt{\log n / \log\log n})$ lower bound of [KMW16, CL21] applies for general n-vertex graphs and $n = \Omega(\sqrt{g})$.

Our contributions. In the remaining of the chapter, by "LOCAL algorithm", we mean deterministic distributed algorithm in the classical LOCAL model.

- We propose a $(34+\varepsilon)$-approximation LOCAL algorithm for MDS in Euler genus-g graphs whose round complexity depends only on g (cf. Corollary III.4.3). This improves approximation ratios in both the orientable (previously $91+\varepsilon$) and non-orientable (previously $24g+O(1)$) genus case. The analysis of the algorithm uses the fact that such graphs have asymptotic dimension at most two.
- This result follows directly from more general meta-theorems. More generally, we prove that any constant-round α-approximation for MDS on planar graphs can be turned into an $f(g)$-round $(3\alpha+1)$-approximation algorithm for MDS on Euler genus-g graphs, for some function f.
- We extend this result in two ways. First, we show that our meta-theorems hold for all local minimization problems that are well-behaved (including MDS and its variants). We provide a simple graph-theoretic property on the problem that is sufficient for our meta-algorithms to apply. This significantly simplifies proofs and circumvents having to reason about various algorithms.
- We apply our results on problems such as Minimum k-Tuple Dominating Set (cf. Corollary III.4.5), obtaining constant-ratio approximations on problems where none were previously known. We also obtain a ratio of $O(\sqrt{g})$ for Minimum Connected Dominating Set in Euler genus-g graphs.
- Second, we prove that our meta-theorems hold on much broader "locally nice" graph classes, such as locally bounded-genus graphs, or locally bounded-genus graphs with a bounded number of apex vertices (whose unconstrained neighborhoods can break the surface embeddability of the graph).
- We prove three versions of our meta-theorem (see Meta-Theorem I, Meta-Theorem II and Meta-Theorem III), making it easily applicable and adapted to different situations.

Unlike similar notions, such as the classical network decomposition [AGLP89], no preliminary construction is required for any of our algorithms. Obviously, they depend on the parameters of the graph classes on which they are applied, like the Euler genus g and/or the asymptotic dimension of the class with its control function.

Organization. Basic notations and asymptotic dimension are introduced in Section III.2. Section III.3 derives a 616-approximation for MDS on bounded-genus graphs, gently introducing the ideas and motivations of the later meta-theorems. Finally, Sections III.4 and III.5 present our different meta-theorems, and their applications to different problems of the literature.

III.2 Preliminaries

Graphs and domination. A subset $X \subseteq V(G)$ *dominates* S if $S \subseteq N[X]$. We denote by $\mathrm{MDS}(G, S)$ the minimum size of a subset of $V(G)$ dominating S. Note that $\mathrm{MDS}(G, V(G))$, denoted by $\mathrm{MDS}(G)$ for short, is the size of a minimum dominating set for G.

Asymptotic dimension. In this section, we focus on defining asymptotic dimension[7] (a non-trivial task, as it happens) and explaining how to exploit it, in the hope that others may be able to exploit it in turn.

A graph G with a spanning subgraph H is *k-colorable δ-bounded in H* if each vertex of G can be assigned a color from $\{0, \ldots, k-1\}$ so that the distance in H between any two vertices taken in a monochromatic path of G is at most δ. In other words, all monochromatic connected components of G must have a weak diameter in H at most δ. A proper k-coloring of G is nothing else than a k-coloring 0-bounded in H. If H has diameter D, then it is 1-colorable D-bounded in H.

The asymptotic dimension of a graph class $\mathscr{C}$ is a non-negative integer denoted by $\mathrm{asdim}(\mathscr{C})$, and related to the k-coloring boundedness of the r-th power of G. It is a notion closely related to *sparse partitions*[8] and *weak sparse covers* (see [BBE⁺24, §1.8] for precise connection between these notions). Recall that the r-th power of G, denoted by G^r, is the graph obtained from $V(G)$ by adding edges between pairs of vertices at distance at most r in G.

Rather than giving its standard definition, we prefer the following equivalent one[9]:

Proposition III.2.1 ([BBE⁺24, Proposition 1.17]). *For every graph class $\mathscr{C}$, $\mathrm{asdim}(\mathscr{C}) \leqslant d$ if and only if there exists a function $f : \mathbb{N} \to \mathbb{N}$, called* control function, *such that for every $G \in \mathscr{C}$ and integer $r \geqslant 1$, G^r is $(d+1)$-colorable $f(r)$-bounded in G.*

Any associated control function f that makes $\mathrm{asdim}(\mathscr{C}) = d$ is also called a d-dimensional control function of $\mathscr{C}$. The Assouad-Nagata dimension of $\mathscr{C}$ has a similar definition, except that the control function must be linear. A crude example is that paths have Assouad-Nagata dimension 1, because, for each $r \geqslant 1$, it suffices to alternatingly color subsets of r consecutive vertices with two colors, see Section III.2.

It is tempting to think of the $k \times k$ grid as a $(k/r) \times (k/r)$ grid of subgrids each of size $r \times r$ and color those subgrids with 4 colors. However, by shifting the subgrids a little, as in Section III.2, one can only use 3 colors instead of 4, proving that the asymptotic dimension of 2-dimensional grids is 2. More generally, d-dimensional grids have a d-dimensional

[7] Asymptotic dimension was originally defined on metric spaces by Gromov [Gro93] in the field of geometric group theory.

[8] However, those notions follow a different philosophy. In the study of asymptotic dimension, we want to minimize the dimension. In the study of sparse partitions and covers, the main goal is to minimize a function of the dimension and the diameter of the sets.

[9] In the original Proposition 1.17 of [BBE⁺24], G^r is $(d+1)$-colorable $f(r)$-bounded in G^r, instead of in G. By following this original definition, the function f is not a control function, whereas in Proposition III.2.1 it is.

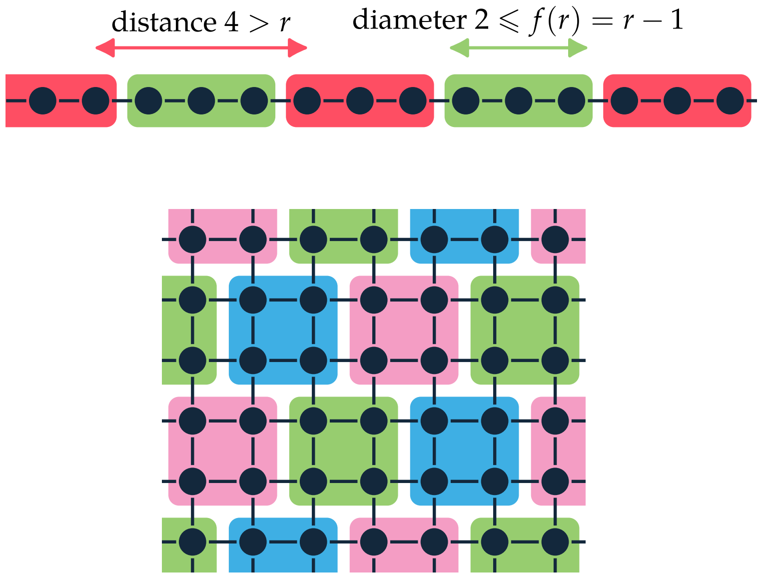


Figure III.1: An example that shows that P^r is 2-colorable $(r-1)$-bounded in P for $r = 3$ (on the top), and an example of a 3-coloring of grids for $r = 2$ and $f(r) = 2(r-1) = 2$ (on the bottom).

control function $f(r) = d \cdot (r-1)$.

Note that any finite graph class $\mathscr{G}$ has asymptotic dimension 0, as one can always take f to be constant, always set to the number of vertices in a largest graph of $\mathscr{G}$, then take one color for the whole graph regardless of r. This highlights an important subtlety: for the asymptotic dimension to have meaning, it needs to be defined on a graph class and not on a single graph.

Trees, and more generally graph classes of bounded treewidth (resp. layered treewidth) have asymptotic dimension 1 (resp. 2). Planar graphs, and more generally the classes of H-minor-free graphs (for any fixed H) have asymptotic dimension 2 as shown by [BBE+24]: the dependency in H only shows in the control function f. Dense graph classes may also have small asymptotic dimension. For example, it is sufficient that there is a quasi-isometry into a class having small asymptotic dimension[10]. The dense class of chordal graphs, being quasi-isometric to the class of trees, has asymptotic dimension 1.

[10] It is conjectured in [BBE+24] that, even more generally, any graph class forbidding some graph as a *fat* minor should also have asymptotic dimension at most 2.

Intuitively, each connected component of a color class in G^r can be viewed as a "simple" component of the graph, that we already understand well. In the setting of LOCAL algorithms with constant round complexity $r/2$, each vertex can observe only its distance-$(r/2)$ neighborhood. If the underlying graph class has asymptotic dimension d, this neighborhood intersects at most $d+1$ such simple parts. As a result, we can derive strong approximation guarantees, since each vertex's view of the graph remains constrained and conceptually simple. We observe that all our algorithms make use of the control function f, but not for any particular pre-computing coloring or partitioning, so that the asymptotic dimension d is actually used only in analysis of the approximation ratio.

III.3 MDS on Bounded Euler Genus Graphs

This section explains how asymptotic dimension helps us devise approximation algorithms for MDS on Euler-genus-g graphs. It also serves as an introduction to the method of cutting and motivates the later meta-theorems. The following algorithm for planar graphs will serve as a black-box building block for the more involved algorithm of Theorem III.3.2.

Proposition III.3.1. *There is a* 205*-approximation* LOCAL *algorithm* A *for* Minimum Dominating Set *in planar graphs with round complexity* 6*.*

The proof of this result can be found in Section III.7.1.

To present the algorithm (cf. Algorithm 1) we first need to define the "best" minimum dominating sets of a labelled graph H. Among all dominating sets, we first discard those which contain some v for which there exists w with $N[v] \subsetneq N[w]$, and among the remaining ones, we declare "best" the one lexicographically smallest considering the labels of the vertices.

Algorithm 1: The algorithm for dominating set on planar graphs.

Require: A planar graph G.
Ensure: A dominating set D of G such that $|D| \leqslant 205 \cdot \mathrm{MDS}(G)$.
1: $D \leftarrow \varnothing$
2: Each vertex u computes $G[N^5[u]]$ and all minimum dominating sets of $N^4[u]$ in G. Among those sets, u selects the "best" one as D_u.
3: Each vertex u picks the vertex v_u of smallest label in $D_u \cap N[u]$, and adds it to D.

By construction, Algorithm 1 outputs a dominating set D within 6 rounds. In fact, Algorithm 1 is very natural in the sense that in order to compute a global dominating set we use local dominating sets. The proof on the approximation factor is given in the Section III.7.1. We extend Algorithm 1 to genus g graphs by dealing separately with local subgraphs of G which are planar and which are not planar. For a graph G, the *T-error set* of G w.r.t. planarity is the vertex set defined by $X = \{u \in V(G) \mid G[N^T[u]] \text{ is not planar}\}$.

Theorem III.3.2. *There exists a function* C *such that for any* g*, there is a* 616*-approximation* LOCAL *algorithm for* Minimum Dominating Set *in Euler genus-*g *graphs with round complexity* $C(g)$*.*

Proof. Suppose G has Euler genus g and let f be a 2-dimensional control function of the class of Euler genus-g graphs. A 2-hop in $X \subset V(G)$ is a sequence of vertices $x_1, \ldots, x_k$ in X such that consecutive vertices are at distance at most two in G.

We run the following algorithm.

Algorithm 2: The algorithm for dominating set on bounded genus graphs.

Require: A graph G of genus $\leqslant g$.
Ensure: A dominating set D of G such that $|D| \leqslant 616 \cdot \mathrm{MDS}(G)$.
1: If $N^{f(10)+5}[u]$ is planar, u runs Algorithm 1.
Let X be the set of undominated vertices after step 1.
2: If $u \in X$, u computes the lexicographically smallest minimum set $D_{X'}$ in G which dominates the set $X' \subseteq X$ of vertices reachable from u with a 2-hop in X. It adds $N[u] \cap D_{X'}$ to D.

Claim III.3.3. *The round complexity is some constant $C(g)$ depending only on g.*

Proof of claim. It suffices to show that there is no 2-hop of weak diameter $\geqslant 2(f(10)+6)g$ of vertices in X. Suppose such a 2-hop on vertices of X exists. Take two vertices x and y on the 2-hop which are at least $2(f(10)+6)g$ apart. Note that for any $i < 2(f(10)+6)g$ there exists a vertex on the 2-hop at distance i or $i+1$ from x. Hence, there exist $u_1, \dots u_{g+1} \in X$ where u_i is at distance $2(f(10)+6)(i-1)$ or $2(f(10)+6)(i-1)+1$ from x. Therefore, the sets $N^{f(10)+5}[u_1], \dots, N^{f(10)+5}[u_{g+1}]$ are pairwise disjoint and they are non-planar, so the genus of the graph is at least $g+1$ by additivity (cf. [MT01, Theorem 4.4.3]). As the genus is smaller than g, this is a contradiction. ◇

The coloring of the asymptotic dimension gives us color classes C_1, C_2 and C_3, such that:

- $C_i = B_1^i \cup \dots \cup B_{j_i}^i$ with[11] $\mathrm{diam}_G(B_j^i) \leqslant f(10)$ for all $1 \leqslant j \leqslant j_i$, and
- for any $j \neq \ell$, we have $B_j^i \neq B_\ell^i$. More precisely, B_j^i and B_ℓ^i are at distance at least 11.

[11] We denote by $\mathrm{diam}_G(X)$ the largest distance in G between any two vertices of X.

The cutting. Let Y_G be a minimum dominating set of G. Let $G' = G[N[Y']]$ where $Y' = Y_G \cap N^5[B_j^i]$. Note that G' contains $N^4[B_j^i]$ and $|\mathrm{MDS}(G')| \leqslant |Y_G \cap N^5[B_j^i]|$.

The analysis outside of the error set. Note that G' either contains only vertices from X or contains at least one vertex which is not in X and is therefore planar. Hence, vertices in $B_j^i \setminus X$ choose at most $205 \cdot |\mathrm{MDS}(G')| \leqslant 205 \cdot |Y_G \cap N^5[B_j^i]|$ vertices as their neighborhood of distance 5 in G and G' is the same. Vertices not in X choose at most $205|Y_G|$ vertices in each color class. Across all color classes, this adds up to at most $3 \cdot 205|Y_G|$.

The analysis with non-empty error set. Let $X_1, \dots, X_k$ be the 2-hop components. Note that for two distinct 2-hop components X_i and X_j, the neighborhoods $N[X_i]$ and $N[X_j]$ do no intersect. As $|D_{X_i}| \leqslant |Y_G \cap N[X_i]|$, the vertices in X choose at most $|Y_G|$ vertices.

In total, at most $3 \cdot 205|Y_G| + |Y_G| = 616 \cdot |\mathrm{MDS}(G)|$ vertices are added to D: the algorithm is a 616-approximation. □

In fact, if Algorithm 1 gives an approximation ratio α, then the above algorithm for genus g graphs is a $(3\alpha + 1)$-approximation algorithm. One might now ask whether any algorithm for MDS on planar graphs can be turned into one for genus g graphs. We show that this is indeed possible. In order to do so, we need to show that the cutting step and its analysis work for any MDS algorithm for planar graphs. We delve into this deeper in the next section.

III.4 Meta-Theorems

In this section, we present our meta-theorems, which take as input a LOCAL approximation algorithm for a graph class $\mathscr{C}$ and returns a LOCAL approximation algorithm for a broader class $\mathscr{D}$ that locally looks like graphs in $\mathscr{C}$, possibly with some local exceptions.

Meta-theorems using black-box LOCAL algorithms might be difficult to state as there are some subtle issues with vertex identifiers (see for instance [FKP13, FGKS13, GHS13]). The LOCAL model assumes the input graph G has unique $O(\log n)$-bit vertex labels, i.e., with identifiers polynomial in G. In particular, running a LOCAL algorithm A on only a small part H of G might fail because: (1) vertex labels in H may not be polynomial in H anymore, and (2) algorithm A may require polynomial-size identifiers to work properly. Thus, Meta-Theorem I assumes that the black-box algorithm A does not require polynomial identifiers. This captures all order-invariant algorithms[12]. In contrast, Meta-Theorem II and Meta-Theorem III allow A to use polynomial identifiers, but impose restrictions on the optimization problem. Meta-Theorem II & III are presented in Section III.5.

[12] A model where vertices have unique labels, but the output of A is invariant under relabelings preserving the label order [GHS13].

From locally-$\mathscr{C}$ to locally nice graphs. Given a graph class $\mathscr{C}$, we say that a graph class $\mathscr{D}$ is *T-locally-$\mathscr{C}$* if for all $G \in \mathscr{D}$ and $u \in V(G)$, $G[N^T[u]] \in \mathscr{C}$.

This definition can already capture many graph classes, such as graphs of girth at least $2k+2$: those are just graphs that are k-locally-$\mathscr{T}$, where $\mathscr{T}$ is the class of trees. Similarly, graphs embedded on a surface with edge-width[13] at least $2k+2$ are k-locally-$\mathscr{P}$, where $\mathscr{P}$ is the class of planar graphs. However, our meta-theorem captures broader classes of graphs as it can support local exceptions.

[13] The length of shortest non-contractible cycle in the embedded graph.

Given a graph class $\mathscr{D}$ and $G \in \mathscr{D}$, the *T-error set* of G w.r.t. $\mathscr{C}$ is the vertex set defined by $X = \{u \in V(G) \mid G[N^T[u]] \notin \mathscr{C}\}$. In other words, this is the set of vertices that are witnesses for G not being T-locally-$\mathscr{C}$. Clearly, if G has no T-errors, that is $X = \varnothing$, then G is T-locally-$\mathscr{C}$. Given a ρ-local problem Π [14], the class $\mathscr{D}$ is T-locally δ-nice w.r.t. $\mathscr{C}$ and Π, if the maximum weak diameter in any $G \in \mathscr{D}$ of any connected component of $G[N^{2\rho}[X]]$ is at most δ, where X is the T-errors of G w.r.t. $\mathscr{C}$. This means that the set of T-errors can be covered by far-away bounded-radius balls of G. By Proposition III.7.11, bounded Euler genus graphs

[14] Which we define formally later.

form a locally nice class of graphs w.r.t. $\mathscr{C}$.

Uniform approximation. Our meta-theorems are inspired by the intriguing property that any approximate algorithm A for MDS is going to perform well in a target graph class $\mathscr{D}$ that is locally-$\mathscr{C}$, as long as $\mathscr{D}$ has small asymptotic dimension, and A makes "good local choices" in $\mathscr{C}$. More precisely, a LOCAL algorithm A is a *k-uniform α-approximation* in $\mathscr{C}$, if for all $G \in \mathscr{C}$ and $S \subseteq V(G)$, $|\mathsf{A}(G) \cap S| \leqslant \alpha \cdot \mathrm{MDS}(G, N^k[S])$, where $\mathsf{A}(G)$ is the set returned by A when run on G, and $\mathrm{MDS}(G, X)$ is the minimum size of a subset of $V(G)$ dominating vertices of X in G. Unless explicitly said, we will assume for convenience that $k \geqslant 1$, as otherwise only very restricted graph classes $\mathscr{C}$ can support 0-uniform constant-approximation algorithms for MDS (cf. Proposition III.8.1).

We can now state properly the fact (proved by us in an earlier paper) that any k-uniform approximate algorithm that performs well for $\mathscr{C}$, performs well for a locally-$\mathscr{C}$ class of bounded asymptotic dimension. It can be shown that the bounded asymptotic dimension assumption cannot be removed, even for MDS[15].

Proposition III.4.1 ([BGPW25, Proposition 3.1])**.** *Let* A *be a k-uniform α-approximation* LOCAL *algorithm for* Minimum Dominating Set *in a hereditary*[16] *class of graphs $\mathscr{C}$ with round complexity $r \geqslant 1$. Let $\mathscr{D}$ be a graph class with d-dimensional control function f, and that is*[17] *$(f(2k+2)+r)$-locally-$\mathscr{C}$. Then* A *is also an $\alpha(d+1)$-approximation algorithm on $\mathscr{D}$.*

We extend Proposition III.4.1 in multiple ways. First, we allow $\mathscr{D}$ to be locally nice, and secondly, we prove that this also applies to problems other than MDS. This means that, for example, any approximate algorithm A solving a well-behaved problem that performs well on planar graphs can be converted into an algorithm B that performs well on bounded genus graphs, as long as A is k-uniform on $\mathscr{C}$, and $\mathscr{D}$ has small asymptotic dimension. Before giving the full statement of Meta-Theorem I, we define the class of problems we are interested in.

LOCAL optimization problems. As defined in [GHS13], a *simple graph problem* Π is an optimization problem in which a feasible solution is a subset of vertices or edges, and the goal is to optimize the size of a feasible solution. A simple graph problem Π is *ρ-local* if there is a LOCAL algorithm R with round complexity ρ that can recognizes a feasible solution[18]. Algorithm R recognizes S as a feasible solution if and only if, whenever applied on G with S given, R returns true for all vertices of G (and thus, at least one vertex should return false if S is not a solution). See Section III.6.1 for more formal definitions. We are mainly interested in small locality problems, that is when ρ is constant, as clearly all simple graph problems are ρ-local for ρ large enough. A lot of well-known problems such as optimization variants of LCL problems [NS95]. This includes Minimum Vertex Cover, Minimum Dominating Set,

[15] This can be achieved by observing that a simple constant-round 3-approximation LOCAL algorithm exists for MDS in trees, but (high) girth-g graphs do not admit a constant-round $o(g)$-factor approximation. The latter follows by combining Ramsey-theoretic arguments with the lower bound proof due to [GHS13] based on $2k$-regular graphs of girth g, where vertices can be linearly ordered so that a positive fraction of them have pairwise isomorphic radius-r neighborhoods, for any r, k, g.

[16] A class of graphs closed under vertex deletion.

[17] The original statement claimed erroneously that $\mathscr{D}$ being t-locally-$\mathscr{C}$ for $t = f(2k+1)$ is enough, instead of the correct $t = f(2k+2) + r$. This has no impact, as in practice we use Proposition III.4.1 in Corollary III.4.4 only for $k \leqslant r = O(1)$.

[18] We do not require that the algorithm checks the optimality of the solution. If we require this, most interesting problems would require $\rho = \Omega(n)$ to be ρ-local.

Maximum Independent Set, Maximal Matching, and Minimum k-Tuple Dominating Set. The distance-d versions of these problems[19] are also d-local problems. However, Minimum Connected Dominating Set is $\Omega(n)$-local[20]. Our framework includes problems such as Red-Blue Dominating Set that can take vertex or edge input labels on the graph, and problems defined on arbitrarily large degree graphs.

In this work, we focus on subsets of vertices, but the techniques presented here extend to subsets of edges. In our setting, we restrict our attention to cases where the vertices do not have access to n, the size of the graph. For some problems, such as Minimum Dominating Set, this restriction can be relaxed[21]. However, we omit this case for brevity, since we are not aware of any constant-round algorithm for a ρ-local problem that requires n. As, in the LOCAL model with constant number of rounds, the literature mostly studies minimization problems, we also focus on those. Observe that many classically interesting problems like Maximum Independent Set and Maximum Matching do not admit constant-time constant-ratio approximations (see [BH25] for references), even on the restricted graph class of paths[22].

Given a ρ-local problem Π, a graph G and $X \subseteq V(G)$, we define $\mathrm{OPT}_\Pi(G, X)$ as the optimal size of a subset of vertices of G that is feasible on every vertex of X (and not necessarily for vertices outside X). We naturally extend the definition of k-uniform algorithm as follows: a LOCAL algorithm A is a *k-uniform* α-approximation for Π in a class of graphs $\mathscr{C}$, if for all $G \in \mathscr{C}$ and $S \subseteq V(G)$, $|\mathsf{A}(G) \cap S| \leqslant \alpha \cdot \mathrm{OPT}_\Pi(G, N^k[S])$.

We can now state our strengthening of Proposition III.4.1. Note that we further require (when the set of errors is not empty) that our problem is *additive*, that is, any superset of a solution feasible on X is still feasible on X. For instance, MDS is additive because adding vertices to any dominating set maintains a dominating set.

Theorem III.4.2 (Meta-Theorem I)**.** *Let* A *be a k-uniform α-approximation* LOCAL *algorithm that does not require polynomial identifiers, for an additive ρ-local problem Π in a class of graphs $\mathscr{C}$ with round complexity r, where $\mathscr{C}$ is hereditary and stable under disjoint union. Let $\mathscr{D}$ be a graph class of asymptotic dimension d with d-dimensional control function f that is T-locally δ-nice w.r.t. $\mathscr{C}$ and Π, where $T = f(2k + 2\rho) + \max\{k + \rho, r\}$. Then, there exists a* $\max\{k, \rho\}$*-uniform $(\alpha(d + 1) + 1)$-approximation algorithm* B *on $\mathscr{D}$ with round complexity $T + \delta + 2$. Furthermore, if $\mathscr{D}$ is T-locally $\mathscr{C}$, the approximation ratio is $\alpha(d + 1)$ and the additivity condition can be dropped.*

The full proof of this result can be found in Section III.7.2. In the following, we present a very rough sketch of the proof instead.

Sketch of proof. The goal of our Meta-Theorem is to transform a local approximation algorithm A that works on a family of graphs $\mathscr{C}$ into one that works on a larger graph class $\mathscr{D}$, with approximation factor multiplied by $d + 1$, where d is the asymptotic dimension of $\mathscr{D}$. We start

[19] E.g. for Minimum Dominating Set, every vertex has a dominating vertex in its distance-d neighborhood.

[20] In an n-vertex cycle, no LOCAL algorithm R of round complexity $< n/2$ can distinguish the situation where a feasible solution is $S = V(G) \setminus \{u\}$ from the situation where $S = V(G) \setminus \{u, v\}$ with two opposite vertices u, v.

[21] In full generality, this applies to all problems for which, for every n, there exists an n-vertex graph within the input graph class that has bounded diameter and admits a constant-size solution. For MDS on planar graphs, we can take a depth-1 tree.

[22] The rough intuition for this is that on paths, any constant-time algorithm run on a vertex far away from the end of the path sees the same neighborhood, and therefore will output the same result on every vertex or edge. The only way to comply with the problem description, unlike minimization problem like MDS, is to not take any vertex in the independent set (or any edge in the maximum matching), at least in the middle of the path. This leads to an arbitrarily bad approximation ratio.

by constructing a LOCAL algorithm B that does the following.

- Run A everywhere, but discard the output of vertices whose local view is not in $\mathscr{C}$. We call this set of vertices the error set.
- Brute-force a local optimal solution of regions around the error set.
- Output the union of the two sets.

This outputs a valid solution to the problem, because the problem is additive: adding a solution for the neighborhood of the error set and a solution for vertices far-away from error set always produces a solution that satisfies every vertex. Now, we prove that this algorithm is indeed a constant-factor approximation. Vertices far from errors are already satisfied by the black-box algorithm. Unsatisfied vertices are near errors, so are satisfied by the brute-force steps, which costs at most the global optimum. Notice that as the class of graphs is locally-nice, the diameter of every errors-set component is bounded. Thus, this brute-force computation step can run in the LOCAL model in constant-time. Let us focus on the other vertices, which are far from the errors. Using asymptotic dimension, we can, given any r, color our input graph with $d+1$ colors. The number of vertices selected by our algorithm is sum over colors c of vertices selected of color c. We bound this number for each color separately. We know that the connected components of each color class have their diameter in G^r bounded with respect to r. By considering the induced subgraph on those components that are far from the error set, we get a graph of our original class $\mathscr{C}$, where our black-box algorithm performs well. Thus, for each component of each color class, we can bound the size of the returned set by uniformity. As those components are far apart, their local neighborhoods do not overlap and their optimal solutions add up independently. For one color, only $\alpha \cdot \mathrm{OPT}$ vertices are taking in the returned set, where OPT is the size of a minimum solution on the whole graph. Summing over all colors and accounting for the brute-force step, we get that only a small number $((\alpha(d+1)+1)\,\mathrm{OPT})$ of vertices are selected in the solution, which proves that the overall returned solution is indeed a good approximation. The running time is constant, as the brute-force step is constant and running the black-box algorithm also takes constant time. □

Cutability. It may be difficult to prove that some algorithm is k-uniform. Instead, we provide a simple sufficient criterion for deciding if *any* constant-round constant-ratio approximation algorithm for Π is k-uniform.

A ρ-local problem Π is *cuttable* for a hereditary graph class $\mathscr{C}$ with parameter $\beta > 0$ if for all $G \in \mathscr{C}$ and $X \subseteq V(G)$, there exists some $Y \supseteq X$ such that $\mathrm{OPT}_\Pi(G[Y]) \leqslant \beta \cdot \mathrm{OPT}_\Pi(G, X)$. Informally, this asks that X can be augmented into a set Y (if not the whole vertex set) so that the graph induced by this bigger set Y has an optimal solution somewhat smaller than the optimal feasible solution for X.

For instance, MDS is cuttable with parameter $\beta = 1$ (see Claim III.7.3), as is the distance-d version of MDS (see Proposition III.7.6). We prove in Section III.7.6 the fundamental property that every constant-round constant-ratio approximation for a cuttable problem is also k-uniform (as long as the algorithm does not require polynomial identifiers). Therefore, MDS admits uniform approximation algorithms. This means we can actually use Meta-Theorem I for any cuttable problem without the k-uniform assumption. Actually, for MDS, we can show that this property holds even if the algorithm requires polynomial identifiers (cf. Proposition III.7.2).

Recall that Euler genus-g graphs have asymptotic dimension at most 2, and are locally nice (cf. Proposition III.7.11). Thus, Meta-Theorem I implies that any constant-round α-approximation for MDS gives a $C(g)$-round $(3\alpha + 1)$-approximation for graphs of Euler genus g.

The best known ratio for MDS in planar graphs is $\alpha = 11 + \varepsilon$ due to Heydt et al. [HKO+25]. Therefore,

Corollary III.4.3. *For any $\varepsilon > 0$, for any g, there is a $k(\varepsilon)$-uniform $(34 + \varepsilon)$-approximation* LOCAL *algorithm for* Minimum Dominating Set *in Euler genus-g graphs with round complexity $C(\varepsilon, g)$ for some functions C and k.*

We can even go one step further by applying Meta-Theorem I to the uniform approximation LOCAL algorithm derived from Corollary III.4.3, and the class $\mathscr{C}$ of Euler genus-g graphs. We immediately obtain the following, where 34.01 can be thought of as $34 + \varepsilon$.

Corollary III.4.4. *Let $\mathscr{C}$ be the class of Euler genus-g plus graphs, and $\mathscr{D}$ be a graph class of asymptotic dimension d that is r-locally-$\mathscr{C}$ for some r large enough. Then, there is a $34.01(d + 1)$-approximation* LOCAL *algorithm for* Minimum Dominating Set *in $\mathscr{D}$ with round complexity $C(d, g, r)$ for some function C.*

Actually, Corollaries III.4.3 and III.4.4 hold for the larger class of Euler genus-g plus a-apex graphs, i.e. graphs obtained from graph of Euler genus g where at most a vertices are allowed to have an arbitrary neighborhood. These graphs exclude $K_{3+a,2g+3+a}$ as minor (as, without apices, the graphs exclude $K_{3,2g+3}$), and thus have asymptotic dimension 2 as well. They are also locally nice, since a apices cannot create short-cuts longer than $2a$. So, the approximation ratio is preserved, whereas the round complexity depends on a.

Moreover, our results can be applied to get approximation algorithms on graphs of bounded Euler genus for *all* other minimization problems in the Distributed Computing literature that admit a constant-round, constant-ratio approximation on planar graphs (for a comprehensive list of such problems, see the survey of Feuilloley [Feu23]).

For instance, let us focus on the Minimum k-Tuple Dominating Set problem (studied extensively in [WAF02]), a generalization of Minimum Dominating Set. In this variant, the goal is to find the smallest subset of vertices such that every vertex outside the set is adjacent to at least

k vertices within it. The classical MDS problem corresponds to the special case $k = 1$. For planar graphs, this problem admits a constant-round $k/(k-2)$-approximation algorithm [CHS⁺14] for $k > 2$, and a 6-approximation [CHS⁺17] for $k = 2$. Both algorithms do not require polynomial identifiers.

Using the fact that MINIMUM k-TUPLE DOMINATING SET is cuttable and additive (cf. Proposition III.7.5), that Euler genus-g graphs are locally nice (cf. Proposition III.7.11), and by applying Proposition III.7.7 and Meta-Theorem I, we get the following:

Corollary III.4.5. *For every $k \geqslant 2$, there is a $C_1(k)$-uniform α-approximation LOCAL algorithm for MINIMUM k-TUPLE DOMINATING SET in Euler genus-g graphs with round complexity $C_2(g,k)$, for some functions C_1, C_2, where $\alpha = 19$ if $k = 2$ and $\alpha \leqslant 3k/(k-2)+1$ otherwise.*

For MINIMUM CONNECTED DOMINATING SET [WAF02], which is an example of non-local problem, we can obtain a constant-round approximation with ratio $O(\sqrt{g})$ in Euler genus-g graphs by applying the "connected" reduction from MDS in bounded expansion graphs due to [AORS18, Theorem 5.8]. This gives an approximation ratio of $2\alpha\nabla_1 \leqslant 68\sqrt{3g/2} + O(1)$, where $\alpha = 34 + \varepsilon$ is the best approximation ratio for MDS in Euler genus-g graphs (Corollary III.4.3), and $\nabla_1 \leqslant \sqrt{3g/2} + 3$ (Proposition III.8.3) is the expansion of Euler genus-g graphs. We leave open the question of improving the ratio to a constant independent of the genus.

For example of non-cuttable problems and why cutability seems unavoidable, at least with our approach, see Section III.6.2.

III.5 When the black-box algorithm requires polynomial identifiers

As we have seen, Proposition III.7.7 applies to lots of optimization problems. However, it does not fulfil the polynomial identifiers requirement of the LOCAL model. We would like to have the same statement to apply in the LOCAL model with polynomial identifiers with the same assumptions. However, assigning new identifiers to vertices while maintaining control of the output of the algorithm is very challenging. Ramsey-type techniques (for example, for finding lower bounds on the approximation ratio on particular graph classes [HLS14]) can be useful for this, but only on very particular graphs: ones where each neighborhood is similar. In general graphs, to the best of our knowledge, we do not know how to adapt this technique. Thankfully, for locally-$\mathscr{C}$ classes, where the set of errors is empty, we can weaken the statement of Proposition III.7.7 in a way that we can still apply a variation of Meta-Theorem I afterwards. The trick is to modify the coloring given by the asymptotic dimension so that all color classes are almost balanced in size (see Lemma III.7.12). With this technique, we just need to apply uniformity on sets that are large. Formally, LOCAL algorithm A is an ε-weakly *k-uniform* α-approximation for Π in a class of graphs $\mathscr{C}$, if for all $G \in \mathscr{C}$ and $S \subseteq V(G)$ such that $|S| \geqslant \lfloor \varepsilon |V(G)| \rfloor$, we have $|\mathsf{A}(G) \cap S| \leqslant \alpha \cdot \mathrm{OPT}_\Pi(G, N^k[S])$. As before, say that the d-local problem Π is *ε-weakly uniformizable* on $\mathscr{C}$ if for every integer $d > 0$ and r and any real α, there exists some integers k, r' and real β such that for every r-round α-approximation A of Π on $\mathscr{C}$ in constant time, there exists some r'-round LOCAL algorithm B that is a ε-weakly k-uniform β-approximation of Π on $\mathscr{C}$ in constant time. The function $(r, \alpha) \mapsto (k, \beta)$ is called the *binding* function.

It is a corollary of the proof of Proposition III.7.7 that any cuttable problem is weakly uniformizable, in the LOCAL model this time (see Corollary III.7.10). Therefore, we can use Lemma III.7.12 to only work on color classes of linear size (which preserves the polynomial-ness of identifiers), and prove the following.

Theorem III.5.1 (Meta-Theorem II)**.** *Let d, k be integers $\alpha > 0$ a real,* A *be a $\frac{1}{d+1}$-weakly k-uniform α-approximation LOCAL algorithm for a ρ-local problem Π in a class of graphs $\mathscr{C}$ with round complexity r, where $\mathscr{C}$ is hereditary and stable by disjoint union. Let $\mathscr{D}$ be a graph class of asymptotic dimension d with d-dimensional control function f, that is T-locally-$\mathscr{C}$ for $T = f(6k + 6\rho) + 4k + 4\rho + \max\{k + \rho, r\}$. Then* A *is a k-uniform $\alpha(d + 1)$-approximation algorithm on $\mathscr{D}$.*

The proof of this can be found in Section III.7.11.

When the error set is not empty, we require further assumptions. For a discussion on why those assumptions are required, we refer to Section III.6.4. A problem Π is called *ω-paddable* if for any integer N, there exists some $G \in \mathscr{C}$ on at least N vertices such that $\mathrm{OPT}_\Pi(G) \leqslant \omega$. We also add a small property that the problem needs to satisfy: a d-local problem Π is *dense* in $\mathscr{C}$ if there exists some unbounded function g such that for any $G \in \mathscr{C}$ of diameter D, we have $\mathrm{OPT}_\Pi(G) \geqslant g(D)$. We require that[23] $\liminf g = +\infty$. This essentially mean that there are no large (in diameter) portion of the graph with no vertices of the solution; every vertex is close to a vertex of the solution. Under both the additional assumptions of paddability and density, we can prove an analogue of Meta-Theorem II. See Meta-Theorem III for the full statement.

Summary table of the results. Here is a simplified table of the assumptions necessary to apply our meta-theorem that transform a constant-round constant-factor approximation algorithm A on the class $\mathscr{C}$ to another constant-round constant-factor approximation algorithm on the class $\mathscr{D}$. We assume that $\mathscr{C}$ is hereditary and stable by disjoint union, and that $\mathscr{D}$ is of bounded asymptotic dimension.

[23] It is possible to get rid of this property if one adds that the graphs in the definition of ω-paddable also have bounded diameter. Then, the rough idea is that, in the proof of Meta-Theorem III, the algorithm will be able to compute the optimal solution on the graph $\widehat{G}$ of the proof. Then one gets Equation (III.6) without the ω term.

[24] It is possible to remove the dense condition, at the expense of making the paddable condition stricter. See Section III.5.

	T-error set is empty ($\mathscr{D}$ is locally-$\mathscr{C}$)	T-error set is not empty ($\mathscr{D}$ is locally nice)
A *does not need* polynomial IDs	Π is cuttable (Meta-Theorem I and Proposition III.7.7)	Π is cuttable and additive (Meta-Theorem I and Proposition III.7.7)
A *needs* polynomial IDs	Π is cuttable (Meta-Theorem II and Corollary III.7.10)	Π is cuttable, additive, paddable and dense[24] (Meta-Theorem III and Corollary III.7.10)

Table III.1: Simplified table of the assumptions necessary to apply Meta-Theorem I, Meta-Theorem II and Meta-Theorem III, along with their references.

III.6 Discussions

III.6.1 Definitions for Locally Verifiable Problems

Let us define ρ-local verifiable problems in more detail. Let G be a graph with a set Σ_{in} of input labels on vertices or edges, which includes a unique identifier for each vertex. If $v \in V(G)$, A is a LOCAL algorithm, and X is a set of vertices of G, we use $\mathsf{A}(G, X, v)$ to represent the output of vertex v when A is run on G with input labels $\Sigma_{\text{in}} \sqcup X$ (the local inputs form an encoding of X, i.e. every vertex knows if it is contained or not in X). A simple graph problem Π is ρ-local if there exists some LOCAL algorithm R which runs in ρ rounds such that for all vertices v and subsets of vertices X, the following holds:

- if X is a feasible solution of the problem Π on (G, Σ_{in}), then for all vertices v of G, we have $\mathsf{A}(G, X, v) = 1$, or
- if X is not a feasible solution of the problem Π on (G, Σ_{in}), then there exists $v \in V(G)$ such that $\mathsf{A}(G, X, v) = 0$.

III.6.2 Why cutability seems unavoidable

Cutability is not a trivial property. Consider the following very artificial problems: we want to output a set (as small as possible) which contains every vertex of degree 0 or 1 in the graph. Of course, this problem admits a trivial 1-round algorithm: take all vertices of degree 0 or 1. However, our goal is to be able to use algorithms in a black-box manner. Instead, we consider the 1-round algorithm A that outputs every vertex of degree not equal to 2. Take $\mathscr{C}$ the class of forests, on which A is a 2-approximation. This is because there are at least as many leaves as vertices of degree 3 or more.

Now, let $\mathscr{D}$ be the closure (under subgraphs) of q-subdivisions of ladders[25], where q is as large as we like, and a ladder is the Cartesian product of an edge and an arbitrarily long path. Clearly, every graph in $\mathscr{D}$ is q-locally in $\mathscr{C}$. Additionally, $\mathscr{D}$ has asymptotic dimension 1, (one can

[25] The q-subdivision of a graph G is a copy of G where every edge is replaced by a path with q inner vertices.

check that they are K_4-minor-free, and so have bounded treewidth hence asymptotic dimension 1 [BBE+24]). However, A fails catastrophically on $\mathscr{D}$. Indeed, there are graphs in $\mathscr{D}$ that do not have any leaves or isolated vertices (one can just take the q-subdivision of a ladder), so that the best solution on this graph is empty. Of course, in this very specific scenario there does exist an alternative algorithm which gives a 1-approx in forests and actually works in $\mathscr{D}$. However, the point is to avoid modifying the black-box A. More generally, such degree-based problems seem to have uncontrollable behavior across local-global boundaries: in $\mathscr{C}$ the algorithm can afford to take more vertices than necessary because of the existence of some other vertices (such as the large amount of leaves in the case of forests), whereas in $\mathscr{D}$ these vertices do not exist.

III.6.3 Cutability of Distance-Exactly-d Dominating Set

In some graph classes such as planar graphs and minor-free classes, Proposition III.7.6 can be extended to Minimum Distance-Exactly-d Dominating Set. Given a graph G and d a positive integer, a subset of vertices D is a called a *distance-exactly-d dominating set* of G if every vertex v of $V(G) \setminus D$ is at distance exactly d from some vertex in D. In the Minimum Distance-Exactly-d Dominating Set problem, one asks for *distance-exactly-d dominating set* of minimum size. In graph classes that are closed under the addition of leaves to the graphs, Distance-$(\leqslant d)$ Dominating Set is cuttable with parameter d. To give a rough intuition, in these classes, one can attach a path of length $d-1$ to a vertex v, put v and all vertices in the path to the dominating set, and thus simulate a distance-$(\leqslant d)$ dominating set that way.

III.6.4 On the additional assumptions of the Meta-Theorem for paddable problems

In Meta-Theorem III, we require problems that are paddable, i.e., we want to exclude graph classes that admit arbitrarily large graphs whose difficulty is concentrated in a bounded-diameter region with bounded optimum. Otherwise, a padding trick can inflate instance size (hence identifier ranges) without increasing hardness. One cannot hope for a version of Meta-Theorem III with non-empty error sets and for problems without this property. Here is the scenario we want to avoid. One could take any approximation algorithm on a class of graph $\mathscr{C}$, and construct a class $\mathscr{D}$ where graphs of size n are formed by taking a graph of $\mathscr{C}$ of size at most $\sqrt{n}$ (or n^c for some $0 < c < 1$), and attaching somewhere an error set X of size at least $n - \sqrt{n}$, which has small radius and small (bounded) optimum solution. Then, this version of Meta-Theorem III would give a good approximation on this class $\mathscr{D}$. However, solving the problem Π efficiently on $\mathscr{D}$ would require solving it on $\mathscr{C}$, as the error set is of small radius and has a small optimum solution. Put in a nutshell, there is roughly a reduction from the class $\mathscr{D}$ to the class $\mathscr{C}$. However, as the error set is large in size, one could attribute very large identifiers to the portion of the graph in $\mathscr{C}$: setting $|X| \geqslant n - \sqrt{n}$ can artificially inflate the identifiers by a square (and one can get a polynomial with power C by choosing $|X| \geqslant n^{1/C}$). If the algorithm does not need the polynomial identifiers, everything still works. If it is not the case however, we have transformed an easy instance to another easy instance where identifiers are larger. This is not possible if the algorithm truly needs the polynomial identifiers, i.e., if the approximation guarantee depends on the vertex identifiers. Thus, we require $\mathscr{C}$ to be paddable or $\mathscr{D}$ to be not paddable. We prove the result for the case where $\mathscr{C}$ is paddable only, as the proof is similar for the other case. [26]

[26] If we require the algorithm to have knowledge the size of the graph, we can have Meta-Theorem III with this additional property. However, to achieve this, we need to change the definition of paddable: we now request, for any n, a graph of size n (instead of size $\geqslant n$) with the desired properties.

III.7 Proofs

III.7.1 Approximation for planar graphs

Proposition III.3.1. *There is a* 205*-approximation LOCAL algorithm* A *for* Minimum Dominating Set *in planar graphs with round complexity* 6.

Proof. Let G be a planar graph, let Y be a nice dominating set of G and let D be the dominating set of G returned by the algorithm. Let $u \in V(G)$ and let D_u be the dominating set computed by u. That means

- D_u is nice
- D_u is the lexicographically smallest nice set dominating $N^4[u]$

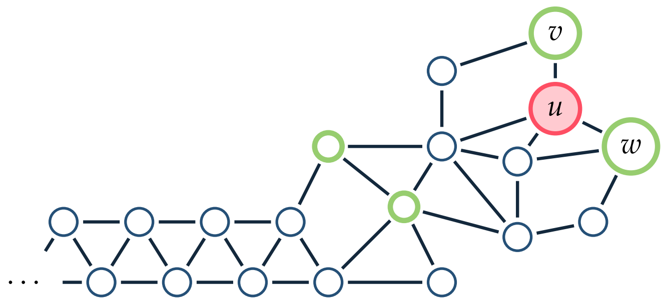


Figure III.2: Depicted in green is the dominating set D_u for u. Note that $d(u)$ is v or w.

The first key observation is that given disjoint subsets A, B, C of $V(G)$ such that $G = G[A \cup C] \cup G[B \cup C]$, that is, C separates A from B (see Figure III.3). If $G[B \cup C]$ is in the fourth neighborhood of $u \in B \cup C$, the value of $D_u \cap B$ is completely determined by $D_u \cap C$. In other words, there is a unique best way to extend $D_u \cap C$ to B. Therefore, for such vertices u, there are at most $2^{|C|} \cdot MDS(G, B \cup C)$ different candidates for $d(u)$.

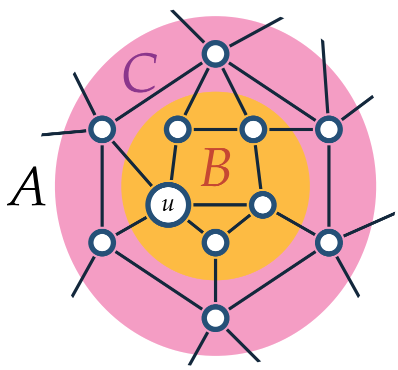


Figure III.3: The depiction of A, B, C in the proof of the key observation.

We now discuss how to use that observation to reach the desired conclusion. We associate to each vertex $u \in V \setminus Y$ an arbitrary vertex $y_u \in Y \cap N(u)$, and we consider the plane multigraph H obtained from an arbitrary planar embedding of G by contracting every edge uy_u for $u \in V \setminus Y$, but keeping any resulting loop or multiple edge. Note that $V(H) = Y$ by construction. Let H' be the plane multigraph obtained from H by iteratively deleting any edge incident to a face bounded by one edge (i.e. a loop) or to two faces bounded by two edges (i.e. there is a multiedge of multiplicity 3), see Section III.7.1. In particular, H' may contain loops and multiple edges, but in the embedding they separate different parts of $V(H')$ or, in the case of multiple edges, are incident on at least one side to a face of size at least 3.

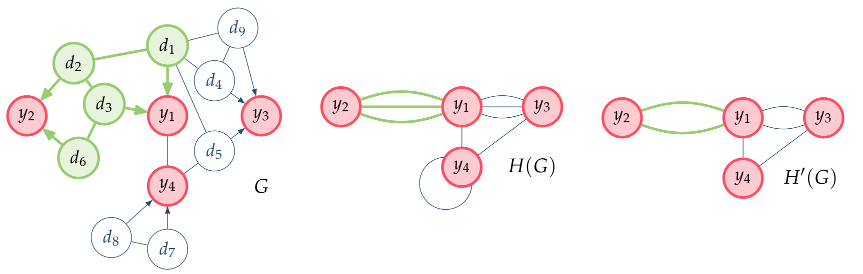


Figure III.4: The plane graphs H and H' obtained from G.

We observe that by Euler's formula, there are fewer than $2 \cdot 3|V(H')| = 6|Y|$ edges in H'. To every edge e of H or H' we associate an edge e_G of G such that e_G turns into e in the contraction process. We let Z be the subset of vertices of $V(G) \setminus Y$ which are an endpoint of e_G for some edge $e \in E(H')$. Since every edge e_G contains at most 2 vertices of Z, we have $|Z| \leqslant 12|Y|$. Therefore, there are at most $13|Y|$ vertices of $Y \cup Z$ in D.

It remains to bound $|D \setminus (Y \cup Z)|$. Note that there are two types of vertices $v \in V(G) \setminus (Y \cup Z)$.

- Either $N[v] \subseteq N[y_v]$, or
- $N[v] \nsubseteq N[y_v]$ and v is between two edges e_G, e'_G in G such that e and e' bound a face of size 2 in H'.

There cannot be any other case, as if the first case does not apply then v has degree at least 2, so is adjacent to some vertex w such that w is not a neighbor of y_u. But then w gets contracted to a vertex $s \neq y_u$ when creating H. Note that vw does not get contracted and let e be the edge in H such that $e_G = vw$. But since v is not in Z, the edge e gets deleted when creating H', i.e. e is in between two parallel edges to e and we are in the second case.

We now assume w is a vertex such that $d(w) = v$. If the first case applies then any neighbor of v is also a neighbor of y_v, in particular, the distance of a vertex $w \neq v$ to v is the same as to y_v. However, either y_v has strictly more neighbors than v, or y_v has a smaller label than v, as $y_v \in Y$, hence w would prefer y_v over v.

We analyze now the second case, which is slightly more interesting. Let e_G be incident to x_1, x_2 and e'_G to w_1, w_2, where for each i, either $d(x_i) = y_i$ or $x_i = y_i$ and similarly for w_i and without loss of generality we assume $y_u = y_1$. For a depiction, see Figure III.5.

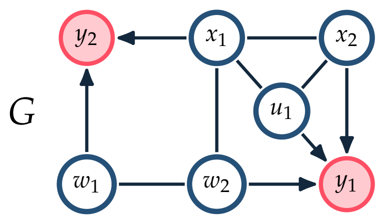


Figure III.5: A depiction of x_1, x_2, w_1, w_2 for the vertex u_1.

Note that $C = \{x_1, x_2, y_1, y_2, w_1, w_2\}$ separates u from the rest of $V(H')$, and that vertices involved in some edge inside f in H are all adjacent to y_1 or to y_2 hence are at distance at most 3 to all vertices in f (with respect to u, since u is a neighbor of y_2 or has a neighbor which is a neighbor of y_2 as we are in the second case). But this means that w is at distance at most 4 to all vertices in f. Therefore, there are at most 2^6 choices for D_u and $2^6 \cdot 2$ choices for $d(u)$ if $d(u) \notin Y \cup Z$. This is because not more than 2 vertices from the set contained in the cycle $y_2 x_1 w_1 y_1 w_2 x_2 y_2$ can be chosen as otherwise y_1, y_2 would be a better choice. We can note that in the case where both y_1 and y_2 get selected in D_u, there is no extra vertex to add, as all of the inside is dominated and when one of y_1 or y_2 is chosen, then there is at most one vertex picked on the inside, otherwise y_1, y_2 would be a better choice. Therefore, we reduce the number of choices to $1 \cdot 2^4 \cdot 2 + 2 \cdot 2^4 \cdot 1 = 64$.

Since there are at most $3|Y|$ options for f, this adds up to $192|Y|$.

All together, we get $13|Y|$ for the set Y or the set Z being chosen, and $192|Y|$ for the choices of the second case of vertices not in Z or Y being chosen. This sums up to $205|Y|$, as claimed. □

III.7.2 *Meta-theorem for the* LOCAL *model with no polynomial identifiers requirement*

Theorem III.4.2 (Meta-Theorem I)**.** *Let* A *be a k-uniform α-approximation* LOCAL *algorithm that does not require polynomial identifiers, for an additive ρ-local problem Π in a class of graphs $\mathscr{C}$ with round complexity r, where $\mathscr{C}$ is hereditary and stable under disjoint union. Let $\mathscr{D}$ be a graph class of asymptotic dimension d with d-dimensional control function f that is T-locally δ-nice w.r.t. $\mathscr{C}$ and Π, where $T = f(2k+2\rho) + \max\{k+\rho, r\}$. Then, there exists a $\max\{k, \rho\}$-uniform $(\alpha(d+1)+1)$-approximation algorithm* B *on $\mathscr{D}$ with round complexity $T + \delta + 2$. Furthermore, if $\mathscr{D}$ is T-locally $\mathscr{C}$, the approximation ratio is $\alpha(d+1)$ and the additivity condition can be dropped.*

This proof is essentially a simplified proof of Meta-Theorem III. However, we include it for completeness.

Proof. Let $G \in \mathscr{D}$. As $\mathscr{D}$ has d-dimensional control function f, then $G^{2k+2\rho}$ admits a $(d+1)$-coloring $f(2k+2\rho)$-bounded in G. Fix such a coloring, and, for each $i \in \{0, 1, \ldots, d\}$, let C_i be the set of color-i vertices in $G^{2k+2\rho}$ (and also in G). We denote by $\mathcal{C}_i$ the set of $(2k+2\rho)$-components of C_i (i.e. connected components of $G^{2k+2\rho}[C_i]$). So, by definition, all components of $\mathcal{C}_i$ have weak diameter in G at most $f(2k+2\rho)$, and are pairwise at distance at least $2k+2\rho+1$ in G (because distinct components of $\mathcal{C}_i$ cannot be adjacent in $G^{2k+2\rho}$).

We describe the Algorithm B we want to use in Algorithm 3, which is very similar to Algorithm 2.

We run Algorithm B on G. It is easy to check that B(G) is a valid

Algorithm 3: Generic algorithm B from Meta-Theorem I.

Require: A k-uniform α-approximate LOCAL algorithm A for Π in $\mathscr{C}$ with round complexity r, graph classes $\mathscr{C}$ and $\mathscr{D}$ with properties of the statement of Meta-Theorem I, the d-dimensional control function f of $\mathscr{D}$ and $G \in \mathscr{D}$ with T-errors X where $T = f(2k+2\rho) + \max\{k+\rho, r\}$.

Ensure: A solution S for the graph G to the problem Π with the guarantee that $|S| \leqslant (\alpha(d+1)+1) \cdot \mathrm{OPT}_\Pi(G)$.

1: $S \leftarrow \varnothing$

2: Each vertex u computes $G[N^T[u]]$ and checks whether it belongs to $\mathscr{C}$; as a consequence, it decides whether it belongs to X.

3: Each vertex u runs A for r rounds, and gets added to S only if $u \in \mathsf{A}(G) \setminus X$.

4: $S \leftarrow S \cup S'$, where S' is a brute-forced minimum set of G that satisfies the constraints of vertices that are non-satisfied.

solution for G, because Π is additive. By definition, $|\mathsf{B}(G)| \leqslant |S| + |S'|$, where $S = \mathsf{A}(G) \setminus X$ (Step 3) and S' is a brute-forced set of minimum size satisfying the constraints of vertices of G not satisfied (Step 4). Note that every vertex not in $N^\rho[X]$ is satisfied by S, so $S' \subseteq N^{2\rho}[X]$. Clearly, $|S'| \leqslant \mathrm{OPT}_\Pi(G)$, and $|S'| = 0$ if there is no T-errors ($X = \varnothing$). Moreover, by definition of the coloring, $|\mathsf{A}(G) \setminus X| = \sum_{i=0}^d |(\mathsf{A}(G) \cap C_i) \setminus X|$. In other words,

$$|\mathsf{B}(G)| \leqslant \sum_{i=0}^{d} |(\mathsf{A}(G) \cap C_i) \setminus X| + \begin{cases} 0 & \text{if } X = \varnothing \\ \mathrm{OPT}_\Pi(G) & \text{otherwise} \end{cases} \tag{III.1}$$

To upper bound the term $|(\mathsf{A}(G) \cap C_i) \setminus X|$, consider some color $i \in \{0, 1, \ldots, d\}$ and some component $C \in \mathcal{C}_i$ such that $C \not\subseteq X$. Consider some $v_C \in C \setminus X$, and denote $G_C = G[N^T[v_C]]$. Note that, by definition of X and as $v \notin X$, we have $G_C \in \mathscr{C}$. Moreover, let $\mathcal{C}'_i = \{C \in \mathcal{C}_i \mid C \not\subseteq X\}$ and let $G' = G[\bigcup_{C \in \mathcal{C}'_i} N^T[v_C]]$ be the union of the G_C's for $C \in \mathcal{C}_i$ such that $C \not\subseteq X$.

Along the proof, we will use several times the following fact:

Fact III.7.1. *For any graph H and $W \subseteq V(H)$, to satisfy the constraints of vertices of W in H, it is enough to select vertices taken from $N^\rho[W]$.*

In the following, to differentiate the neighborhoods in G and in G_C or G', we write them using either N_G, N_{G_C}, or $N_{G'}$ notations. Notice that, for each non-negative integer $t \leqslant \max\{k+\rho, r\}$, we have $G[N_G^t[C]] \subseteq G_C$. This is because C has weak diameter in G at most $f(2k+2\rho)$ and thus $G[N_G^t[C]]$ has weak diameter in G at most $f(2k+2\rho) + t \leqslant T$, by the choice of T. Moreover, as $\mathscr{C}$ is hereditary, we have that $G[N_G^t[C]] \in \mathscr{C}$.

As $G[N_G^t[C]] \subseteq G_C$, vertices in C have the same distance t-neighborhood in G and G_C. In particular, $G[N_G^t[C]] = G_C[N_{G_C}^t[C]]$ and $G[N_G^t[\bigcup \mathcal{C}'_i]] = G'[N_{G'}^t[\bigcup \mathcal{C}'_i]]$. From Fact III.7.1, any minimum set of vertices of G that satisfies the constraints of $N_G^k[C]$ is contained in $G[N_G^{k+\rho}[C]]$, and similarly if we replace G by G_C. It follows that $\mathrm{OPT}_\Pi(G, N_G^k[C]) =$

$\mathrm{OPT}_\Pi(G_C, N^k_{G_C}[C])$. Recall that any two connected components $C, C' \in \mathcal{C}_i$ are at distance in G at least $2k+2\rho+1$. So $G[N_G^{k+\rho}[C]]$ and $G[N_G^{k+\rho}[C']]$ are disjoint subgraphs, and, as to satisfy all constraints of vertices in $V(G)$, one needs to satisfy constraints of vertices in $N^k_G[C]$ and $N^k_G[C']$, by disjunction of these sets and by Fact III.7.1, we have by summing over all $C \in \mathcal{C}_i$ that

$$\mathrm{OPT}_\Pi\left(G, N^k_G\left[\bigcup \mathcal{C}'_i\right]\right) = \mathrm{OPT}_\Pi\left(G', N^k_{G'}\left[\bigcup \mathcal{C}'_i\right]\right).$$

Moreover, because $\mathscr{C}$ is stable by disjoint union and that $G[N^t_G[C]] \in \mathscr{C}$ for every $C \in \mathcal{C}'_i$, we get $G[N^t_G[\bigcup \mathcal{C}'_i]] \in \mathscr{C}$. Therefore, as A is a k-uniform α-approximation on $\mathscr{C}$, we get

$$\left|\mathsf{A}(G') \cap \bigcup \mathcal{C}'_i\right| \leqslant \alpha \cdot \mathrm{OPT}_\Pi\left(G', N^k_{G'}\left[\bigcup \mathcal{C}'_i\right]\right) = \mathrm{OPT}_\Pi\left(G, N^k_G\left[\bigcup \mathcal{C}'_i\right]\right).$$

Since $G[N^r_G[\bigcup \mathcal{C}'_i]] = G'[N^r_{G'}[\bigcup \mathcal{C}'_i]]$, Algorithm A, which runs in r rounds, returns the same vertex set in G and G' for vertices in $\bigcup \mathcal{C}'_i$. Therefore, $|\mathsf{A}(G) \cap \bigcup \mathcal{C}'_i| = |\mathsf{A}(G') \cap \bigcup \mathcal{C}'_i|$. Combining with the previous inequality, we get

$$\begin{aligned}\left|\mathsf{A}(G) \cap \bigcup \mathcal{C}'_i\right| \;\leqslant\; \alpha \cdot \mathrm{OPT}_\Pi\left(G, N^k_G\left[\bigcup \mathcal{C}'_i\right]\right) &\leqslant \alpha \cdot \mathrm{OPT}_\Pi(G, V(G)) \\ &= \alpha \cdot \mathrm{OPT}_\Pi(G)\ .\end{aligned}$$

We remark that $|(\mathsf{A}(G) \cap C) \setminus X| = 0$ if $C \subseteq X$. Because all vertices in $\mathcal{C}_i$ but not in X are in some $C \in \mathcal{C}'_i$, we have

$$|(\mathsf{A}(G) \cap \mathcal{C}_i) \setminus X| \leqslant \left|\mathsf{A}(G) \cap \bigcup \mathcal{C}'_i\right| \leqslant \alpha \cdot \mathrm{OPT}_\Pi(G)\ . \tag{III.2}$$

Putting Equation (III.2) in Equation (III.1), we get that Algorithm B is an $(\alpha(d+1)+1)$-approximation, and even an $\alpha(d+1)$-approximation if $X = \varnothing$.

To see that B is also a k-uniform approximation, it is sufficient to see that, for any $W \subseteq V(G)$, the bound obtained is $|\mathsf{B}(G) \cap W| \leqslant \alpha(d+1) \cdot \mathrm{OPT}_\Pi(G, N^k_G[W]) + \mathrm{OPT}_\Pi(G, N_G[W])$ where the first two terms are due to running A and the last term is due to running the brute-force. Indeed, if the brute-force computed a set whose intersection with W was smaller than $\mathrm{OPT}_\Pi(G, N^\rho_G[W])$, we could replace it by the minimum size set satisfying the constraints of the vertices in $N^\rho_G[W]$ and obtain a smaller set, a contradiction. This proves that B is a $\max\{k,\rho\}$-uniform with an approximate ratio $\alpha(d+1)+1$ (or $\alpha(d+1)$ if $X = \varnothing$, as the brute-force is not required in that case). Now, let us prove that Algorithm B has the desired round complexity.

- Computing X in Step 2 takes $T+1$ rounds.
- Running A in Step 3 takes r rounds.
- For Step 4, consider the set S computed by B at Step 3, before the brute-force. Observe that the vertices in the set $W = V(G) \setminus N^\rho_G[X]$

are satisfied by S. This is because vertices of W are at distance at least $\rho+1$ from X and thus vertices of $N_G^{\rho}[W]$ cannot be in X. Thus, A applies to all vertices of $N_G^{\rho}[W]$, and so W is indeed satisfied by S (Fact III.7.1). Therefore, to satisfy $N_G^{\rho}[X]$ it is enough to select a set S' from $N_G^{2\rho}[X]$ (Fact III.7.1) as done in Step 4. By assumption, the connected components of $N_G^{2\rho}[X]$ have weak diameter in G at most δ (obviously, if $X=\varnothing$ then $\delta=0$). Therefore, the brute-force (Step 5) will take at most $\delta+1$ rounds.

Naively, Steps 2 and 3 together take $(T+1)+r$ rounds. As Algorithm A is not guaranteed to work when executed on vertices of X, Step 3 must be run only after Step 2. However, both steps can be run in parallel as follows. We run A for r rounds exactly and stop it just after (since its running time on a vertex of X could result into more than r rounds). Then, the decision to add u in S (if selected by A) is delayed for the next $T+1-r$ rounds (this is non-negative from the choice of T). In parallel, $G[N^T[u]]$ is computed and is checked to be in $\mathscr{C}$ or not after $T+1$ rounds. So, after $\max\{r, T+1\} = T+1$ rounds, set S in Step 3 has been completed. Step 4 takes $\delta+1$ steps, so that the total round complexity of B is $T+\delta+2$, completing the proof. □

III.7.3 Minimum Dominating Set is uniformizable

Proposition III.7.2. *Let $\mathscr{C}$ be a hereditary class of unbounded degree[27] and that is closed by disjoint union. Then, the Minimum Dominating Set problem is uniformizable in the LOCAL model with binding function $(r,\alpha)\mapsto(r+1,\alpha+\varepsilon)$ for any real $\varepsilon>0$.*

[27] For a graph class in which every graph has maximum degree Δ, the LOCAL algorithm that includes all nodes is a $\Delta+1$-approximation. This implies that the bounded-degree case is less interesting than the unbounded-degree one, which is therefore the focus of our study.

Proof. We first prove this statement in the LOCAL model with $\varepsilon=0$. Assume for sake of contradiction that there exists integers d, r and a real α such that for all integers k and real β, there exists an r-round α-approximation A of Π in constant time on the hereditary graph class $\mathscr{C}$ such that A is not k-uniform with β-approximation of Π in constant time. Here, we will take $k=r+1$ and $\beta=\alpha$. By the definition of uniformity, this means that there exists some subset S of vertices for which:

$$|\mathsf{A}(G)\cap S| > \alpha\cdot \mathrm{MDS}(G, N^k[S])\ .$$

Let us cut G around S: set $G' := G[N^k[S]\cup X]$ where $X\subseteq N^{k+1}[S]$ is a minimum dominating set of $N^k[S]$ in G. We keep the same identifiers in G' as in G. We first bound the size of the minimum dominating set of G'.

Claim III.7.3. $\mathrm{MDS}(G')\leqslant \mathrm{MDS}(G,N^k[S])$.

Proof of claim. By definition of X, it follows that $|X|=\mathrm{MDS}(G,N^k[S])$. X is also a dominating set of G', because each of its vertices is either in X

or in $N^k[S]$, and that in the second case, it is dominated by some vertex of X by definition of X. Therefore, $\mathrm{MDS}(G') \leqslant |X| = \mathrm{MDS}(G, N^k[S])$. ◇

As our goal is to get a contradiction on $G' \in \mathscr{C}$ [28] instead of G, we prove the following.

Claim III.7.4. *On S, algorithm A returns the same set in G and in G'.*

This will prove that $|\mathsf{A}(G) \cap S| = |\mathsf{A}(G') \cap S|$.

Proof of claim. A is an r-round algorithm. Therefore, vertices in S only see vertices and edges in $G[N^r[S]]$, and edges between vertices of $N^r(S)$ and vertices of $N^{r+1}(S)$ [29]. All of those are the same in G and G', therefore $\mathsf{A}(G) \cap S = \mathsf{A}(G') \cap S$, as wanted. ◇

By Claim III.7.4,

$$|\mathsf{A}(G')| \geqslant |\mathsf{A}(G') \cap S| = |A(G) \cap S| > \alpha\, \mathrm{MDS}(G, N^k[S])\ .$$

By Claim III.7.3, we have $\alpha\, \mathrm{MDS}(G, N^k[S]) \geqslant \alpha \cdot \mathrm{MDS}(G')$. It follows that $|\mathsf{A}(G')| \geqslant \alpha \cdot \mathrm{MDS}(G')$, a contradiction as $G' \in \mathscr{C}$ and A is an α-approximation on $\mathscr{C}$. Let us now adapt this proof for the LOCAL model, for $\varepsilon > 0$. For now, by taking $\gamma = \alpha + \varepsilon$, we have proved $|\mathsf{A}(G')| \geqslant (\alpha + \varepsilon) \cdot \mathrm{MDS}(G')$. The only thing we need to take care of is that nodes in G have identifiers that are bounded by a polynomial of $|V(G')|$. Instead of changing the identifiers in G' without impacting the output of A, which is tricky, we consider a bigger graph. Let N be the biggest identifier in G'. $\mathscr{C}$ is a hereditary graph class of unbounded degree, so there exists a graph $H \in \mathscr{C}$ and a vertex v of H with degree at least N.[30] Take $H' = H[N_H[v]]$ and give the vertices identifiers in $[N+1, 2N]$. Let $G'' = G' \sqcup H'$. Clearly, H' can be dominated by choosing the single vertex v, i.e. $\mathrm{MDS}(G'') = \mathrm{MDS}(G') + 1$. Furthermore, $\mathsf{A}(G'') = \mathsf{A}(G') \sqcup \mathsf{A}(H')$ and it follows

$$|\mathsf{A}(G'')| \geqslant (\alpha + \varepsilon) \cdot \mathrm{MDS}(G') + 1 = (\alpha + \varepsilon) \cdot \mathrm{MDS}(G'') + 1 - (\alpha + \varepsilon)\ .$$

Now, we can apply the same trick as in Proposition III.8.2. Observe that the diameter of each connected component of the graph G' is at most $3\,\mathrm{MDS}(G')$. We construct an algorithm B where each vertex checks whether the diameter D of its component is less than $3B$, for $B = (\alpha - 1)/\varepsilon$. This can be done by collecting its radius-$3B$ neighborhood in $3B$ rounds. If true, the vertex locally brute forces an optimal solution for its component, which it already knows about. Otherwise, the vertex applies the approximate algorithm A. If $D \leqslant B$, then the algorithm is obviously a good approximation (with approximation ratio 1). Otherwise, $D > 3B$ and it follows that $\mathrm{MDS}(G') > B$. One can verify that $|\mathsf{A}(G'')| > \alpha \cdot \mathrm{MDS}(G'')$, a contradiction, as $G'' \in \mathscr{C}$. □

[28] because $\mathscr{C}$ is hereditary

[29] The definition of an r-round LOCAL algorithm running on vertex v includes edges at distance $r+1$ of v but not vertices at distance $r+1$ of v in G.

[30] Here, if the algorithm requires access to n (the size of the graph), one needs to choose $N \geqslant n$ and delete neighbors of v so that, in the following, we get $|V(G'')| = n$.

III.7.4 Minimum k-Tuple Dominating Set is cuttable and additive

Proposition III.7.5. *Minimum k-Tuple Dominating Set is additive and cuttable with parameter* 1.

Proof. By definition, the problem is additive. Let us now prove cutability. Let G be a graph and $X \subseteq V(G)$, and let S be a k-Tuple Dominating Set of the set X. Set $Y = X \cup S$. S is a k-Tuple Dominating Set of $G[Y]$. Indeed, every vertex of $G[Y]$ that is in $Y \setminus S$ is in $X \setminus S$, and therefore is k-Tuple Dominated by S. Thus, for $\Pi =$ Minimum k-Tuple Dominating Set, we have $\mathrm{OPT}_\Pi(G, X) \geqslant \mathrm{OPT}_\Pi(G[Y])$. □

III.7.5 Distance-($\leqslant d$) Dominating Set is cuttable

Proposition III.7.6. *Distance-($\leqslant d$) Dominating Set is cuttable with parameter* 1.

Proof. Let Π be the Distance-($\leqslant d$) Dominating Set problem, let G be a graph and X a subset of vertices. Let D be a minimum-size set that is distance-d dominating for the vertices in X, so that $|D| = \mathrm{OPT}_\Pi(G, X)$. Note that $D \subseteq N^d[X]$. We are going to construct a set $X \subseteq Y \subseteq N^d[X]$ such $\mathrm{OPT}_\Pi(G[Y]) \leqslant \mathrm{OPT}_\Pi(G, X)$. We do this recursively. Start with $Y := X$. We maintain the invariant that every vertex of Y that is not dominated by $D \cap Y$ in $G[Y]$ is in X. This invariant holds trivially for $Y = X$. While some vertex v of X is not dominated in $G[Y]$ by $D \cap Y$, do the following. Let $y \in D$ be a vertex that dominates v in G, i.e., with the distance to y at most d. Note that here, we do not have necessarily $y \notin Y$: it could be the case that v is dominated by some $y \in X$ in G because of a path with a portion outside X. As $y \in D \subseteq N^d[X]$, there is a path P in $G[N^d[X]]$ from v to y with length at most d. Actually, one can just take the shortest vy-path. It has length at most d and is in $N^d[v] \subseteq N^d[X]$. Set $Y := Y \cup V(P)$. All vertices that are in P (including v) are dominated by y, and therefore are now dominated in $G[Y]$. The invariant is thus satisfied. Each time we do this operation, at least one more vertex of X is dominated in $G[Y]$. As X is finite, this process terminates and we have constructed a set Y that is dominated by $D \cap Y$ in $G[Y]$. It follows that $\mathrm{OPT}_\Pi(G[Y]) \leqslant \mathrm{OPT}_\Pi(G, X)$, and this completes the proof. □

III.7.6 Cutting implies uniform algorithms in the LOCAL model with no polynomial identifiers requirement

A problem Π is *uniformizable* on $\mathscr{C}$ if for all integers r and all reals α, there exists some integers k and real β such that every r-round α-approximation A of Π on $\mathscr{C}$ is a k-uniform β-approximation of Π o, $\mathscr{C}$. The function $(r, \alpha) \mapsto (k, \beta)$ is called the *binding* function.

Proposition III.7.7. *Let $\mathscr{C}$ be a hereditary class that is closed by disjoint union, and let Π be a ρ-local problem which is cuttable with parameter β. Then, Π is uniformizable* in the LOCAL model with no polynomial identifiers requirement *with binding function* $(r, \alpha) \mapsto (r + 1, \alpha \cdot \beta)$.

Proof. Assume for sake of contradiction that there exist an integer r and a real α such that for all integers k, r' and real γ, there exists an r-round α-approximation A of Π on the hereditary graph class $\mathscr{C}$ but no r'-round k-uniform γ-approximation algorithm B for Π on $\mathscr{C}$. Here, we take $k = r + 1$, $\gamma = \alpha \cdot \beta$, $r' = r$ and B = A, i.e. suppose A is not a k-uniform γ-approximation of Π. By the definition of uniformity, this means that there exists some subset S of vertices for which:

$$|\mathsf{A}(G) \cap S| > \alpha\beta \cdot \mathrm{OPT}_\Pi(G, N^k[S]).$$

As Π is cuttable with the particular case $X = N^k[S]$, there exists a set Y such that $\mathrm{OPT}_\Pi(G[Y]) \leqslant \beta \cdot \mathrm{OPT}_\Pi(G, N^k[S])$. We now study the graph $G' := G[Y]$, with the same identifiers as in G. It follows:

Claim III.7.8. *We have* $\mathsf{A}(G) \cap S = \mathsf{A}(G') \cap S$.

Proof of claim. This is the same proof as in Claim III.7.4, except with a set Y instead. As A is an r-round algorithm, vertices in S only see vertices and edges in $G[N^r[S]]$, and edges between vertices of $N^r(S)$ and vertices of $N^{r+1}(S)$. Those are the same in G and G' as $N^{r+1}[S] = N^k[S] \subseteq Y$, and so we have $\mathsf{A}(G) \cap S = \mathsf{A}(G') \cap S$, as wanted. ◇

By Claim III.7.8,

$$|\mathsf{A}(G')| \geqslant |\mathsf{A}(G') \cap S| = |A(G) \cap S| > \alpha\beta \cdot \mathrm{OPT}_\Pi(G, N^k[S]).$$

By the cutability, we have $\mathrm{OPT}_\Pi(G') \leqslant \beta \cdot \mathrm{OPT}_\Pi(G, N^k[S])$. It follows that $|\mathsf{A}(G')| > \alpha \cdot \mathrm{OPT}_\Pi(G')$, a contradiction as $G' \in \mathscr{C}$ and A is an α-approximation on $\mathscr{C}$. □

III.7.7 *The algorithm of Heydt et al. is uniform*

Observation III.7.9. *The algorithm described in [HKO+25, Theorem 2.3] is a $k(\varepsilon)$-uniform $(11 + \varepsilon)$-approximation LOCAL algorithm for Minimum Dominating Set in planar graphs with round complexity $C(\varepsilon)$, for every $\varepsilon > 0$, and for some functions C and k.*

Proof. Let us consider the algorithm described in [HKO+25, Theorem 2.3]. It has a number of rounds bounded by a function of ε, and does not need polynomial identifiers so Observation III.7.9 follows by Proposition III.7.2. □

For the interested reader, here is a more detailed analysis of the running time of the algorithm. It returns a solution $D_1 \cup D_2 \cup D_3^1 \cup D_3^2$. D_1 can be computed in two rounds, D_2 in two additional rounds, and

D_3^1 in one more additional round. The set D_3^2 can be computed in time $O(\log \Delta / \log(1+\varepsilon))$ where $\Delta = 4\nabla_1 \cdot (4^{\nabla_1} + 2\nabla_1)(\Delta_R + 1)/\varepsilon$ with (with the particular case of planar graphs) $\nabla_1 < 3$ and $\Delta_R = \kappa^{s-1}(t + s - 1 + (s-1)\kappa^3)$ with $\kappa = \max\{2\nabla_0, 2\nabla\}$. Again, for planar graphs, $\nabla_0 < 3$ and $\nabla \leqslant 2$ so $\kappa \leqslant 6$. It follows that $\Delta_R \leqslant 15732$ because $s = t = 3$ on planar graphs. So $\Delta \leqslant 13593312/\varepsilon = O(1/\varepsilon)$ and the running time is $O(\log(1/\varepsilon)/\log(1+\varepsilon/2)) = O(\log(1/\varepsilon)/\varepsilon)$.

III.7.8 *Cutability implies weakly uniform algorithms*

Corollary III.7.10. *Let $\mathscr{C}$ be a hereditary class closed by disjoint union, and let Π be a ρ-local problem which is cuttable with parameter β. Then, for any real $\varepsilon > 0$, the problem Π is ε-weakly uniformizable* in the LOCAL model *with binding function $(r, \alpha) \mapsto (r+1, \alpha \cdot \beta)$.*

Proof. Let us go through the proof of Proposition III.7.7 one more time. For sake of brevity, we do not copy the whole identical proof, but instead give a summary of the proof. Assume for sake of contradiction that there exists integers d, r and a real α such that for all integers k and real γ, there exists an r-round α-approximation A of Π in constant time on the hereditary graph class $\mathscr{C}$ such that A is not ε-weakly k-uniform with γ-approximation of Π in constant time. Here, we will take $k = r + 1$ and $\gamma = \alpha \cdot \beta$. By the definition of weak uniformity, this means that there exists some subset $S \subseteq V(G)$ of size at least $\varepsilon|V(G)|$ for which:

$$|\mathsf{A}(G) \cap S| > \alpha\beta \cdot \mathrm{OPT}_\Pi(G, N^k[S]) .$$

We have showed there exists a set Y such that $\beta \cdot \mathrm{OPT}_\Pi(G, N^k[S]) \geqslant \mathrm{OPT}_\Pi(G[Y])$. Let $G' := G[Y]$. We keep the same identifiers in G' as in G. We have showed that $|\mathsf{A}(G')| \geqslant \alpha \cdot \mathrm{OPT}_\Pi(G')$, which would be a contradiction with the fact that α-approximation on $\mathscr{C}$, if A was an algorithm in the LOCAL model with no polynomial identifiers requirement. For this to be a contradiction in the LOCAL model, we have to show that the IDs in G' are also polynomial in $|V(G')|$. Now, we just use the additional property that S has large size. All the identifiers in G' are polynomial in $|V(G)|$, say smaller than $D \cdot |V(G)|^c$ for some real $c > 0$. As $|V(G')| \geqslant |S| \geqslant \varepsilon \cdot |V(G)|$, the identifier in G' are smaller than $D \cdot |V(G')|^c/\varepsilon^c$, which is still polynomial. This finishes the proof. $\square$

III.7.9 *Bounded Euler genus graphs are locally nice*

Proposition III.7.11. *Let Π be a ρ-local problem. Then, for any $g \geqslant 1$, the class of Euler genus-g graphs is T-locally δ-nice w.r.t. planar graphs and Π, where $\delta < g \cdot (2T + 4\rho + 1)$.*

Proof. To derive a contradiction, consider a connected component H of $G[N^{2\rho}[X]]$, and assume that H has weak diameter $\delta \geqslant g \cdot (2T + 4\rho + 1)$ in G. As $g \geqslant 1$, X and H are not empty, and H is not planar. So,

H must contain some path P such that the distance in G between its endpoints, say s to t, is δ. In particular, for each $d \in \{0, 1, \ldots, \delta\}$, there must exist a vertex of P that is at distance exactly d in G from s. This is because the distance in G from s, when moving along an edge of P, is a function that can vary by at most one unit, and this distance goes from 0 (at s) to δ (at t). For every $i \in \{0, 1, \ldots, g\}$, let u_i be any vertex of P at distance exactly $d_i = i \cdot (2T + 4\rho + 1)$ in G from s. So $u_0 = s$, $u_g = t$, and all the u_i's exist in P (so in H) since $d_i \in \{0, 1, \ldots, \delta\}$. By definition of H, each u_i intersects $N^{2\rho}[X]$. So, for each u_i one can select a vertex $x_i \in X$ at distance in G at most 2ρ from u_i. By the triangle inequality, for all $0 \leqslant i < j \leqslant g$, x_i and x_j are at distance in G at least $(d_j - 2\rho) - (d_i + 2\rho) = (j - i) \cdot (2T + 4\rho + 1) - 4\rho \geqslant 2T + 1$. It follows that $N^T[x_i]$ and $N^T[x_j]$ are disjoint. By definition of X and H, the subgraphs $G[N^T[x_i]]$'s are not planar, thus of Euler genus $\geqslant 1$. By additivity of the Euler genera[31] of these $g + 1$ pairwise disjoint subgraphs of G (cf. [MT01, Theorem 4.4.3]), we get a contradiction that G has Euler genus g. Thus, $\delta < g \cdot (2T + 4\rho + 1)$ as claimed. □

[31] The plural of genus. Latin is hard to *handle*.

III.7.10 *Balanced asymptotic dimension*

Lemma III.7.12. *Let $\mathcal{G}$ be a graph class with asymptotic dimension at most d and control function f. For every graph $G \in \mathcal{G}$ with n vertices and every integer $r \geqslant 1$, there exists a cover $C_0, C_1, \ldots, C_d$ of $V(G)$ such that*

- *every r-component of each C_i has weak diameter in G at most $f(3r) + 2r$, and*
- *for every $i \in \{0, \ldots, d\}$ we have $|C_i| \geqslant \lfloor \frac{n}{d+1} \rfloor$.*

Proof. Let $q := \lfloor \frac{n}{d+1} \rfloor$ and r be any integer. By Proposition III.2.1 applied to scale $3r$, there exists a $(d+1)$-coloring of G^{3r} whose monochromatic components have weak diameter at most $f(3r)$ (as they have weak diameter $f(3r)$ in G^{3r}) in G. Let $C_0, C_1, \ldots, C_d$ be the corresponding color classes, with $|C_0| \leqslant |C_1| \leqslant \cdots \leqslant |C_d|$ without loss of generality. They form a cover of $V(G)$, and every $3r$-component of each C_i has weak diameter at most $f(3r)$ in G.

We construct sets $C'_0, \ldots, C'_d$ by proving the following by induction of i. For every i, there exists sets $C'_0, \ldots, C'_{i-1}$ such that

(I1) for every $j \leqslant i$, we have $|C'_j| \geqslant q$, and if $|C'_j| > q$ then $C'_j = C_j$,

(I2) every r-component of each C'_j (for $j \leqslant i$) has weak diameter at most $f(3r) + 2r$ in G, and

(I3) $\bigcup_{j \leqslant i} C'_j \supseteq \bigcup_{j \leqslant i} C_j$.

Note that (I3) says that the sets $C'_0, \ldots, C'_i$ cover at least every vertex of $C_0, \ldots, C_i$. For $i = 0$ these conditions are trivially satisfied. Assume the conditions hold for some $i \in \{0, \ldots, d\}$. Let $A := N^r[C_i] \setminus C_i$ and

$B := (V(G) \setminus N^r[C_i]) \setminus C_i$, such that $A \sqcup B = V(G) \setminus C_i$. Intuitively, we will try to add vertices to U from A whenever possible (these are within distance r of the original C_i), and otherwise take vertices from B (which are farther from C_i).

If $|C_i| \geqslant q$, then take $C'_j = C_j$ for all $j \geqslant i$. (I1) is satisfied because $q \leqslant |C_i| \leqslant \cdots \leqslant |C_d|$. (I2) is satisfied because the C_i's have all weak diameter at most $f(3r)$, and (I3) is trivially satisfied. So assume $|C_i| < q$ and let $s := q - |C_i| > 0$ be the number of new vertices we must add to C_i to reach size q. There are two possibilities.

- Case 1: $|A| \geqslant s$. Choose an arbitrary set $X \subseteq A$ of size s and let $C'_i := C_i \cup X$; this satisfies (I1) and (I3). Let us prove (I2). Let $x, x' \in A$ be two vertices in the same r-component of C'_i. As $C'_i \subseteq N^r[C_i]$, there exist $y, y' \in C_i$ such that $d(x,y), d(x',y') \leqslant r$. We first prove that y and y' are in the same $3r$-component of C_i. Because x and x' are in the same r-component of C'_i, exists a sequence $x = x_1, x_2, \ldots, x_t = x'$ in C'_i with $d(x_j, x_{j+1}) \leqslant r$ for all j. As $C'_i \subseteq N^r[C_i]$, we can choose, for each j, some $y_j \in C_i$ such that $d(x_j, y_j) \leqslant r$. Without loss of generality, choose $y_1 := y$ and $y_t := y'$. Then for any j, we have that

$$d(y_j, y_{j+1}) \leqslant d(y_j, x_j) + d(x_j, x_{j+1}) + d(x_{j+1}, y_{j+1}) \leqslant 3r,$$

  and so y and y' are in the same $3r$-component of C_i, which means that $d(y, y') \leqslant f(3r)$. Hence,

$$\begin{aligned} d(x,x') &\leqslant d(x,y) + d(y,y') + d(y',x') \\ &\leqslant r + f(3r) + r = f(3r) + 2r, \end{aligned}$$

  as required. This proves (I2).

- Case 2: $|A| < s$. Then by definition of B,

$$|B| \geqslant |V(G)| - |A| - |C_i| \geqslant n - s + 1 - |C_i| = n - q + 1.$$

  As the sets C_j for $j \neq i$ cover B, there exists some $j \neq i$ such that $|C_j \cap B| \geqslant (n - q + 1)/d \geqslant q$. Choose an arbitrary set $X \subseteq C_j \cap B$ of size s and let $C'_i := C_i \cup X$; this satisfies (I1) and (I3). (I2) is true all r-components of C'_i are either r-components of C_i (so they have diameter at most $f(3r)$) or are subsets of r-components of C_j (and the same diameter bound applies).

In both cases we have constructed sets C'_i that satisfies (I1), (I2) and (I3). By (I3), $\bigcup_{i=0}^d C'_i = \bigcup_{i=0}^d C_i = V(G)$, so the C'_i form a cover. (I1) and (I2) prove the two wanted conditions on the cover. This finishes the proof. □

III.7.11 *Meta-theorem for non-paddable problems with empty error set*

Theorem III.5.1 (Meta-Theorem II). *Let d, k be integers $\alpha > 0$ a real,* A *be a $\frac{1}{d+1}$-weakly k-uniform α-approximation* LOCAL *algorithm for a ρ-local problem Π in a class of graphs $\mathscr{C}$ with round complexity r, where $\mathscr{C}$ is hereditary and stable by disjoint union. Let $\mathscr{D}$ be a graph class of asymptotic dimension d with d-dimensional control function f, that is T-locally-$\mathscr{C}$ for $T = f(6k+6\rho) + 4k + 4\rho + \max\{k+\rho, r\}$. Then* A *is a k-uniform $\alpha(d+1)$-approximation algorithm on $\mathscr{D}$.*

Proof. Let $G \in \mathscr{D}$ and set $\ell := f(6k+6\rho) + 4k + 4\rho$. As $\mathscr{D}$ has d-dimensional control function f, then by Lemma III.7.12, one can assume $G^{2k+2\rho}$ admits a $(d+1)$-coloring ℓ-bounded in G where every color class has size at least $\lfloor \frac{n}{d+1} \rfloor$. Fix such a coloring, and, for each $i \in \{0, 1, \ldots, d\}$, let C_i be the set of color-i vertices in $G^{2k+2\rho}$ (and also in G), so that $|C_i| \geqslant \lfloor \frac{n}{d+1} \rfloor$. We denote by $\mathcal{C}_i$ the set of $(2k+2\rho)$-components of C_i (i.e. connected components of $G^{2k+2\rho}[C_i]$). So, by definition, all components of $\mathcal{C}_i$ have weak diameter in G at most ℓ, and are pairwise at distance at least $2k+2\rho+1$ in G (because distinct components of $\mathcal{C}_i$ cannot be adjacent in $G^{2k+2\rho}$).

We run Algorithm A on G. It is easy to check that $\mathsf{A}(G)$ is a valid solution for G, because $\mathscr{D}$ is T-locally-$\mathscr{C}$. By definition of the coloring,

$$|\mathsf{A}(G)| = \sum_{i=0}^{d} |\mathsf{A}(G) \cap C_i| \ . \tag{III.3}$$

To upper bound $|\mathsf{A}(G) \cap C_i|$, consider some color $i \in \{0, \ldots, d\}$ and some component $C \in \mathcal{C}_i$. Consider some $v_C \in C$, and denote $G_C = G[N^T[v_C]]$. Note that as $\mathscr{D}$ is T-locally-$\mathscr{C}$, we have $G_C \in \mathscr{C}$. Let $G' = G[\bigcup_{C \in \mathcal{C}_i} N^T[v_C]]$ be the union of the G_C's for $C \in \mathcal{C}_i$.

Along the proof, we will use several times the following fact:

Fact III.7.1. *For any graph H and $W \subseteq V(H)$, to satisfy the constraints of vertices of W in H, it is enough to select vertices taken from $N^\rho[W]$.*

In the following, to differentiate the neighborhoods in G and in G_C or G', we write them using either N_G, N_{G_C}, or $N_{G'}$ notations. Notice that, for each non-negative integer $t \leqslant \max\{k+\rho, r\}$, we have $G[N_G^t[C]] \subseteq G_C$. This is because C has weak diameter in G at most $\ell = f(6k+6\rho) + 4k + 4\rho$ and thus $G[N_G^t[C]]$ has weak diameter in G at most $f(6k+6\rho) + 4k + 4\rho + t \leqslant T$, by the choice of T. Moreover, as $\mathscr{C}$ is hereditary, we have that $G[N_G^t[C]] \in \mathscr{C}$.

Because $G[N_G^t[C]] \subseteq G_C$, vertices in C have the same distance t-neighborhood in G and G_C. In particular, $G[N_G^t[C]] = G_C[N_{G_C}^t[C]]$ and $G[N_G^t[C_i]] = G'[N_{G'}^t[C_i]]$. From Fact III.7.1, any minimum set of vertices of G that satisfies the constraints of $N_G^k[C]$ is contained in $G[N_G^{k+\rho}[C]]$, and similarly if we replace G by G_C. We have $\mathrm{OPT}_\Pi(G, N_G^k[C]) =$

$\mathrm{OPT}_\Pi(G_C, N^k_{G_C}[C])$. Recall that any two connected components $C, C' \in \mathcal{C}_i$ are at distance in G at least $2k+2\rho+1$. So $G[N_G^{k+\rho}[C]]$ and $G[N_G^{k+\rho}[C']]$ are disjoint subgraphs, and, as to satisfy all constraints of vertices in $V(G)$, one needs to satisfy constraints of vertices in $N^k_G[C]$ and $N^k_G[C']$, by disjunction of these sets and by Fact III.7.1, we have by summing over all $C \in \mathcal{C}_i$ that

$$\mathrm{OPT}_\Pi(G, N^k_G[C_i]) = \mathrm{OPT}_\Pi(G', N^k_{G'}[C_i]).$$

Moreover, because $\mathscr{C}$ is stable by disjoint union and that $G[N^t_G[C]] \in \mathscr{C}$ for every $C \in \mathcal{C}'_i$, we get $G[N^t_G[C_i]] \in \mathscr{C}$. As Algorithm A is a $\frac{1}{d+1}$-weakly k-uniform α-approximation on $\mathscr{C}$, that $G' \in \mathscr{C}$ and that $|C_i| \geqslant \lfloor \frac{|V(G)|}{d+1} \rfloor \geqslant \lfloor \frac{|V(G')|}{d+1} \rfloor$ (and therefore G' indeed has polynomial identifiers), we get by definition of weak uniformity that

$$|\mathsf{A}(G') \cap C_i| \leqslant \alpha \cdot \mathrm{OPT}_\Pi(G', N^k_{G'}[C_i]) = \alpha \cdot \mathrm{OPT}_\Pi(G, N^k_G[C_i]) .$$

Since $G[N^r_G[C_i]] = G'[N^r_{G'}[C_i]]$, Algorithm A, which runs in r rounds, returns the same vertex set in G and G' for vertices in C_i. Therefore, $|\mathsf{A}(G) \cap C_i| = |\mathsf{A}(G') \cap C_i|$. Combining with the previous inequality, we get

$$\begin{aligned} |\mathsf{A}(G) \cap C_i| \leqslant \alpha \cdot \mathrm{OPT}_\Pi(G, N^k_G[C_i]) &\leqslant \alpha \cdot \mathrm{OPT}_\Pi(G, V(G)) \\ &= \alpha \cdot \mathrm{OPT}_\Pi(G) . \end{aligned} \tag{III.4}$$

Putting Equation (III.4) in Equation (III.3), we get that Algorithm A is an $\alpha(d+1)$-approximation on $\mathscr{D}$. □

III.7.12 *Meta-theorem for paddable problems with non-empty error set*

Theorem III.7.13 (Meta-Theorem III). *Let d, k be integers $\alpha > 0$ a real, A be a ε-weakly* [32] *k-uniform α-approximation LOCAL algorithm for an additive ρ-local problem Π in a class of graphs $\mathscr{C}$ with round complexity r, where $\mathscr{C}$ is hereditary, stable by disjoint union, and ω-paddable for Π. Let $\mathscr{D}$ be a graph class of asymptotic dimension d with d-dimensional control function f such that Π is dense in $\mathscr{D}$, and such that $\mathscr{D}$ is T-locally δ-nice w.r.t. $\mathscr{C}$ and Π, where $T = f(2k+2\rho) + \max\{k+\rho, r\}$. Then there exists a function g such that for any $\eta > 0$, there exists a $\max\{k,\rho\}$-uniform $(\alpha(d+1)+1+\eta)$-approximation algorithm B on $\mathscr{D}$ with round complexity $\max\{T+\delta+2, g(\eta)\}$.*

[32] Note that, for the proof, we could also prove as a preliminary that weakly uniform and paddable implies uniform. However, for sake of brevity, we prove everything directly in the proof.

Proof. Let $G \in \mathscr{D}$ and X be the set of T-errors of G with respect to $\mathscr{C}$. As $\mathscr{D}$ has d-dimensional control function f, then $G^{2k+2\rho}$ admits a $(d+1)$-coloring $f(2k+2\rho)$-bounded in G. Fix such a coloring, and, for each $i \in \{0, 1, \ldots, d\}$, let C_i be the set of color-i vertices in $G^{2k+2\rho}$ (and also in G). We denote by $\mathcal{C}_i$ the set of $(2k+2\rho)$-components of C_i (i.e. connected components of $G^{2k+2\rho}[C_i]$). So, by definition, all components of $\mathcal{C}_i$ have weak diameter in G at most $f(2k+2\rho)$, and are pairwise at distance at least $2k+2\rho+1$ in G (because distinct components of $\mathcal{C}_i$ cannot be adjacent in $G^{2k+2\rho}$).

We describe the Algorithm B we want to use in Algorithm 4, which is very similar to Algorithm 2.

Algorithm 4: Generic algorithm B from Meta-Theorem III.

Require: An ε-weakly k-uniform α-approximate LOCAL algorithm A for Π in $\mathscr{C}$ with round complexity r, a graph classes $\mathscr{C}$ and $\mathscr{D}$ with properties of the statement of Meta-Theorem III, the d-dimensional control function f of $\mathscr{D}$ and $G \in \mathscr{D}$ with T-errors X where $T = f(2k+2\rho) + \max\{k+\rho, r\}$.

Ensure: A solution S for the graph G to the problem Π with the guarantee that $|S| \leqslant (\alpha(d+1)+1) \cdot \mathrm{OPT}_\Pi(G)$.

1: Let B be such that any graph G of $\mathscr{D}$ with $\mathrm{diam}(G) \geqslant B$ has $\mathrm{OPT}_\Pi(G) \geqslant \alpha(d+1)\omega/\eta$. If $\mathrm{diam}(G) < B$, brute-force an optimal solution. $\triangleright$ B exists as Π is dense in $\mathscr{D}$.
2: $S \leftarrow \varnothing$
3: Each vertex u computes $G[N^T[u]]$ and checks whether it belongs to $\mathscr{C}$; as a consequence, it decides whether it belongs to X.
4: Each vertex u runs A for r rounds, and gets added to S only if $u \in \mathsf{A}(G) \setminus X$.
5: $S \leftarrow S \cup S'$, where S' is a brute-forced minimum set of G that satisfies the constraints of vertices that are non-satisfied.

We run Algorithm B on G. It is easy to check that $\mathsf{B}(G)$ is a valid solution for G, because Π is additive. By definition, $|\mathsf{B}(G)| \leqslant |S| + |S'|$, where $S = \mathsf{A}(G) \setminus X$ (Step 4) and S' is a brute-forced set of minimum size satisfying the constraints of vertices of G not satisfied (Step 5). Note that every vertex not in $N^\rho[X]$ is satisfied by S, so $S' \subseteq N^{2\rho}[X]$. Clearly, $|S'| \leqslant \mathrm{OPT}_\Pi(G)$, and $|S'| = 0$ if there is no T-errors ($X = \varnothing$). Moreover, by definition of the coloring, $|\mathsf{A}(G) \setminus X| = \sum_{i=0}^{d} |(\mathsf{A}(G) \cap C_i) \setminus X|$. In other words,

$$|\mathsf{B}(G)| \leqslant \sum_{i=0}^{d} |(\mathsf{A}(G) \cap C_i) \setminus X| + \begin{cases} 0 & \text{if } X = \varnothing \\ \mathrm{OPT}_\Pi(G) & \text{otherwise} \end{cases} \tag{III.5}$$

To upper bound the term $|(\mathsf{A}(G) \cap C_i) \setminus X|$, consider some color $i \in \{0, 1, \ldots, d\}$ and some component $C \in \mathcal{C}_i$ such that $C \not\subseteq X$. Consider some $v_C \in C \setminus X$, and denote $G_C = G[N^T[v_C]]$. Note that, by definition of X and as $v \notin X$, we have $G_C \in \mathscr{C}$. Moreover, let $\mathcal{C}'_i = \{C \in \mathcal{C}_i \mid C \not\subseteq X\}$ and let $G' = G[\bigcup_{C \in \mathcal{C}'_i} N^T[v_C]]$ be the union of the G_C's for $C \in \mathcal{C}_i$ such that $C \not\subseteq X$.

Along the proof, we will use several times the following fact:

Fact III.7.1. *For any graph H and $W \subseteq V(H)$, to satisfy the constraints of vertices of W in H, it is enough to select vertices taken from $N^\rho[W]$.*

In the following, to differentiate the neighborhoods in G and in G_C or G', we write them using either N_G, N_{G_C}, or $N_{G'}$ notations. Notice that, for each non-negative integer $t \leqslant \max\{k+\rho, r\}$, we have $G[N_G^t[C]] \subseteq G_C$. This is because C has weak diameter in G at most $f(2k+2\rho)$ and thus $G[N_G^t[C]]$ has weak diameter in G at most $f(2k+2\rho)+t \leqslant T$, by the choice of T. Moreover, as $\mathscr{C}$ is hereditary, we have that $G[N_G^t[C]] \in \mathscr{C}$.

Because $G[N_G^t[C]] \subseteq G_C$, vertices in C have the same distance t-neighborhood in G and G_C. In particular, $G[N_G^t[C]] = G_C[N_{G_C}^t[C]]$ and $G[N_G^t[\bigcup \mathcal{C}_i']] = G'[N_{G'}^t[\bigcup \mathcal{C}_i']]$. From Fact III.7.1, any minimum set of vertices of G that satisfies the constraints of $N_G^k[C]$ is contained in $G[N_G^{k+\rho}[C]]$, and similarly if we replace G by G_C. It follows that $\mathrm{OPT}_\Pi(G, N_G^k[C]) = \mathrm{OPT}_\Pi(G_C, N_{G_C}^k[C])$. Recall that any two connected components $C, C' \in \mathcal{C}_i$ are at distance in G at least $2k+2\rho+1$. So $G[N_G^{k+\rho}[C]]$ and $G[N_G^{k+\rho}[C']]$ are disjoint subgraphs, and, as to satisfy all constraints of vertices in $V(G)$, one needs to satisfy constraints of vertices in $N_G^k[C]$ and $N_G^k[C']$, by disjunction of these sets and by Fact III.7.1, we have by summing over all $C \in \mathcal{C}_i$ that

$$\mathrm{OPT}_\Pi\left(G, N_G^k\left[\bigcup \mathcal{C}_i'\right]\right) = \mathrm{OPT}_\Pi\left(G', N_{G'}^k\left[\bigcup \mathcal{C}_i'\right]\right).$$

Moreover, because $\mathscr{C}$ is stable by disjoint union and that $G[N_G^t[C]] \in \mathscr{C}$ for every $C \in \mathcal{C}_i'$, we get $G[N_G^t[\bigcup \mathcal{C}_i']] \in \mathscr{C}$.

In the following, we will choose a graph $\widehat{G}$ with size linear in $|V(G)|$ so that adding $\widehat{G}$ to any subgraph H of G' will form a graph with size linear in $|V(\widehat{G} \sqcup G)|$. This serves two purposes: we are now able to apply weak uniformity on $\widehat{G} \sqcup H$, and we can attribute unique identifiers to $\widehat{G}$ so that the identifiers of $\widehat{G} \sqcup H$ are polynomial (we keep the same identifiers for H). Let $N := \lceil \varepsilon |V(G)|/(1-\varepsilon) \rceil$. By paddability, there exists a graph $\widehat{G} \in \mathscr{C}$ on at least N vertices such that $\mathrm{OPT}_\Pi(\widehat{G}) \leqslant \omega$. Let I be the largest identifier of G. We give identifiers to vertices of $\widehat{G}$ from the set $\{I+1, I+2, \ldots, I+|V(G')|\}$. Let G'' be the disjoint union of G' and $\widehat{G}$ and let $Y = \bigcup \mathcal{C}_i' \sqcup V(\widehat{G})$. The identifiers of vertices in G'' are indeed polynomial in the size of G'', as $\widehat{G}$ has size at least linear in $|V(G)|$. Our goal now is to apply weak uniformity in G'' to the set Y. Note that we only care about $|\mathsf{A}(G') \cap \bigcup \mathcal{C}_i'|$ but need Y to apply weak uniformity. We first prove that $|Y| \geqslant \varepsilon |V(G'')|$.

$$\begin{aligned}|Y| &\geqslant |V(\widehat{G})| \geqslant \varepsilon|V(\widehat{G})| + (1-\varepsilon)|V(\widehat{G})| \\ &\geqslant \varepsilon|V(\widehat{G})| + (1-\varepsilon)N \geqslant \varepsilon|V(\widehat{G})| + \varepsilon|V(G)| \geqslant \varepsilon|V(G'')|\,.\end{aligned}$$

Therefore, by ε-weak k-uniformity α-approximation of A on $\mathscr{C}$, we get

$$\left|\mathsf{A}(G'') \cap Y\right| \leqslant \alpha \cdot \mathrm{OPT}_\Pi(G'', N^k[Y])\,.$$

Notice that

$$|\mathsf{A}(G'') \cap Y| = |\mathsf{A}(G') \cap \bigcup \mathcal{C}_i'| + |\mathsf{A}(\widehat{G})|$$

and that

$$\begin{aligned}\mathrm{OPT}_\Pi(G'', N_{G''}^k[Y]) &= \mathrm{OPT}_\Pi(\widehat{G}) + \mathrm{OPT}_\Pi(G', N_{G'}^k[\bigcup \mathcal{C}_i']) \\ &\leqslant \omega + \mathrm{OPT}_\Pi(G', N_{G'}^k[\bigcup \mathcal{C}_i'])\end{aligned}$$

because $\widehat{G}$ and G' are in disjoint components of G''. Forgetting about the term $|\mathsf{A}(\widehat{G})|$, it follows that

$$\left|\mathsf{A}(G') \cap \bigcup \mathcal{C}_i'\right| \leqslant \alpha \cdot \omega + \alpha \cdot \mathrm{OPT}_\Pi \left(G', N^k_{G'}\left[\bigcup \mathcal{C}_i'\right]\right) . \tag{III.6}$$

Since $G[N^r_G[\bigcup \mathcal{C}_i']] = G'[N^r_{G'}[\bigcup \mathcal{C}_i']]$, Algorithm A, which runs in r rounds, returns the same vertex set in G and G' for vertices in $\bigcup \mathcal{C}_i'$. Therefore, $|\mathsf{A}(G) \cap \bigcup \mathcal{C}_i'| = |\mathsf{A}(G') \cap \bigcup \mathcal{C}_i'|$. Combining with the previous inequality, we get

$$\begin{aligned}\left|\mathsf{A}(G) \cap \bigcup \mathcal{C}_i'\right| &\leqslant \alpha \cdot \omega + \alpha \cdot \mathrm{OPT}_\Pi \left(G, N^k_G\left[\bigcup \mathcal{C}_i'\right]\right) \\ &\leqslant \alpha \cdot \omega + \alpha \cdot \mathrm{OPT}_\Pi(G, V(G)) \\ &= \alpha \cdot \omega + \alpha \cdot \mathrm{OPT}_\Pi(G) .\end{aligned}$$

We remark that $|(\mathsf{A}(G) \cap C) \setminus X| = 0$ if $C \subseteq X$. Because all vertices in C_i but not in X are in some $C \in \mathcal{C}_i'$, we have

$$|(\mathsf{A}(G) \cap C_i) \setminus X| \leqslant \left|\mathsf{A}(G) \cap \bigcup \mathcal{C}_i'\right| \leqslant \alpha \cdot \omega + \alpha \cdot \mathrm{OPT}_\Pi(G) . \tag{III.7}$$

Now, we can apply the same trick as in Proposition III.8.2. As Π is dense in $\mathscr{D}$, there exists some B such that any graph of $\mathscr{D}$ of diameter at least B has optimum value at least $\alpha(d+1)\omega/\eta$. Algorithm B checks (Step 1) whether the diameter D of its component is less than B. This can be done by collecting the radius-B neighborhood of every vertex in B rounds. If true, the vertex locally brute-forces an optimal solution for its component, which it already knows about. Otherwise, the vertex continues the algorithm with Step 2. If $D < B$, then the algorithm is obviously a good approximation (with approximation ratio 1). Otherwise, $D \geqslant B$ and it follows that $\mathrm{OPT}_\Pi(G) \geqslant \alpha(d+1)\omega/\eta$. Putting in Equation (III.7) in Equation (III.5), we get that Algorithm B is an $(\alpha(d+1)+1+\eta)$-approximation, and even an $(\alpha(d+1)+\eta)$-approximation if $X = \varnothing$.

To see that B is also a k-uniform approximation, it is sufficient to see that, for any $W \subseteq V(G)$, the bound obtained is $|\mathsf{B}(G) \cap W| \leqslant \alpha(d+1)\omega + \alpha(d+1) \cdot \mathrm{OPT}_\Pi(G, N^k_G[W]) + \mathrm{OPT}_\Pi(G, N_G[W])$ where the first two terms are due to running A and the last term is due to running the brute-force. Indeed, if the brute-force computed a set whose intersection with W was smaller than $\mathrm{OPT}_\Pi(G, N^\rho_G[W])$, we could replace it by the minimum size set satisfying the constraints of the vertices in $N^\rho_G[W]$ and obtain a smaller set, a contradiction. This proves that B is $\max\{k,\rho\}$-uniform with approximate ratio $\alpha(d+1)+1$ (or $\alpha(d+1)$ if $X = \varnothing$, as the brute-force is not required in that case). Now, let us prove that Algorithm B has the desired round complexity.

- Step 1 takes $B = g(\eta)$ rounds for computing if the diameter is small, and no additional round for the brute-force.
- Computing X in Step 3 takes $T+1$ rounds.

- Running A in Step 4 takes r rounds.
- For Step 5, consider the set S computed by B at Step 4, before the brute-force. Observe that the vertices in the set $W = V(G) \setminus N_G^\rho[X]$ are satisfied by S. This is because vertices of W are at distance at least $\rho + 1$ from X and thus vertices of $N_G^\rho[W]$ cannot be in X. Thus, A applies to all vertices of $N_G^\rho[W]$, and so W is indeed satisfied by S (Fact III.7.1). Therefore, to satisfy $N_G^\rho[X]$ it is enough to select a set S' from $N_G^{2\rho}[X]$ (Fact III.7.1) as done in Step 5. By assumption, the connected components of $N_G^{2\rho}[X]$ have weak diameter in G at most δ (obviously, if $X = \varnothing$ then $\delta = 0$). Therefore, the brute-force (Step 5) will take at most $\delta + 1$ rounds.

Naively, Steps 3 and 4 together take $(T + 1) + r$ rounds. As Algorithm A is not guaranteed to work when executed on vertices of X, Step 4 must be run only after Step 3. However, both steps can be run in parallel as follows. We run A for r rounds exactly and stop it just after (since its running time on a vertex of X could result into more than r rounds). Then, the decision to add u in S (if selected by A) is delayed for the next $T + 1 - r$ rounds (this is non-negative from the choice of T). In parallel, $G[N^T[u]]$ is computed and is checked to be in $\mathscr{C}$ or not after $T + 1$ rounds. So, after $\max\{r, T + 1\} = T + 1$ rounds, set S in Step 4 has been completed. The same trick can be applied for Step 1. Step 5 takes $\delta + 1$ steps, so that the total round complexity of B is $\max\{T + \delta + 2, g(\eta)\}$, completing the proof. $\square$

III.8 Miscellaneous propositions

Proposition III.8.1. *Every graph class $\mathscr{C}$ including trees has no 0-uniform α-approximations for* Minimum Dominating Set*, for every constant α.*

Proof. Consider a depth-2 tree T_α, with $\alpha + 1$ vertices at depth-1, each having $\alpha^2 + 1$ neighbors at depth-2. See Section III.8 for an example. Clearly, $\mathrm{MDS}(T_\alpha) = \alpha + 1$. Moreover, any α-approximation algorithm A in T_α has to select all depth-1 vertices. If not, each of the t non-selected vertex at depth 1 would force the selection of $\alpha^2 + 1$ extra vertices at depth 2, creating by this way a solution with at least $(\alpha + 1 - t) + t \cdot (\alpha^2 + 1) = \alpha \cdot (t\alpha + 1) + 1$ vertices. For $t \geqslant 1$, this is at least $\alpha \cdot (\alpha + 1) + 1 > \alpha \cdot \mathrm{MDS}(T_\alpha)$. However, for a subset S composed of all depth-1 vertices of T_α (so with $|S| = \alpha + 1$), we have on one side $|\mathsf{A}(T_\alpha) \cap S| = |S| = \alpha + 1$, whereas $\mathrm{MDS}(T_\alpha, N^0[S]) = \mathrm{MDS}(T_\alpha, S) = 1$ (considering the root). So $|\mathsf{A}(T_\alpha) \cap S| \nleqslant \alpha \cdot \mathrm{MDS}(T_\alpha, N^0[S])$. Therefore, A cannot be 0-uniform in T_α. $\square$

Proposition III.8.2. *For every $\varepsilon > 0$, any r-round algorithm that returns a dominating set of size at most $\alpha \cdot \mathrm{MDS}(G) + \beta$ on G, can be converted*

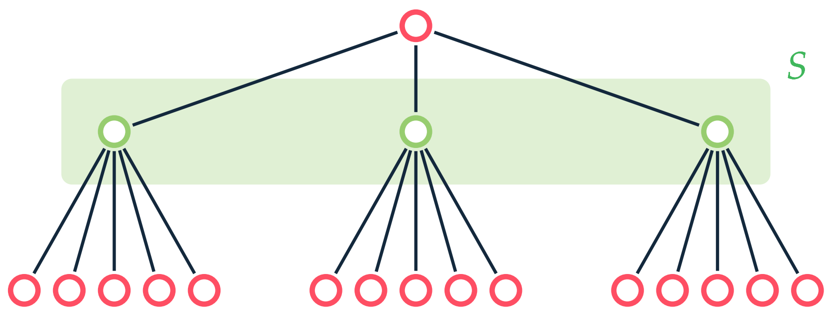


Figure III.6: The tree T_α has no 0-uniform α-approximations (here with $\alpha = 2$). Indeed, any α-approximation A of its minimum dominating set must contain the $\alpha+1$ vertices of S (green), and thus $|\mathsf{A}(T_\alpha) \cap S| = |S| \nleqslant \alpha \cdot \mathrm{MDS}(T_\alpha, N^0[S])$ since S can be dominated by a single vertex (the root).

into an $(\alpha+\varepsilon)$-approximation LOCAL *algorithm with round complexity $r + O(\beta/\varepsilon)$.*

Proof. Observe that the diameter of each connected component of the graph is at most $3 \cdot \mathrm{MDS}(G)$. So, each vertex can check whether the diameter D of its component is less than B, for $B = 3\beta/\varepsilon$. This can be done by collecting its radius-B neighborhood in $B = O(\beta/\varepsilon)$ rounds. If true, the vertex locally brute forces an optimal solution for its component that, which it already knows about. If false, the vertex applies the approximate algorithm. In that case, $B \leqslant D$ and $D \leqslant 3\,\mathrm{MDS}(G)$, which together imply $\beta \leqslant \varepsilon \cdot \mathrm{MDS}(G)$. Thus, the returned set has size at most $\alpha \cdot \mathrm{MDS}(G) + \beta \leqslant (\alpha+\varepsilon) \cdot \mathrm{MDS}(G)$. ☐

Proposition III.8.3. *Every n-vertex graph G of Euler genus g satisfies $\nabla_1(G) < \sqrt{3g/2} + 3$. This bound is best possible up to an additive constant, for each $g \geqslant 0$.*

Proof. By definition, we have $\nabla_1(G) = \max_H \{|E(H)| \,/\, |V(H)|\}$, where H is a depth-1 minor. For every H of genus g, we have $|E(H)| \,/\, |V(H)| < 3g/\,|V(H)| + 3$ from Euler's formula. As $|E(H)| \leqslant \binom{|V(H)|}{2}$, we have

$$\nabla_1(G) \leqslant \max_{m \in [1,n]} \min\{(m-1)/2, 3g/m + 3\} < \sqrt{3g/2} + 3.$$

The latter inequality holds, because either $3g/m + 3 < \sqrt{3g/2} + 3$, and we are done, or $3g/m + 3 \geqslant \sqrt{3g/2} + 3$, which implies $m \leqslant \sqrt{6g}$. In that case $(m-1)/2 \leqslant (\sqrt{6g}\,)/2 - 1/2 < \sqrt{3g/2} + 3$ as wanted.

Consider a graph G composed of a clique K_m with $m \geqslant 4$ and $m \neq 7$ where each edge is replaced by a path of length three. The graph G has same Euler genus as K_m, which is $g = \lceil (m-3)(m-4)/6 \rceil$ (cf. [MT01, Theorem 4.4.5]). Moreover, it has K_m as depth-1 minor. Thus, $\nabla_1(G) \geqslant |E(K_m)| \,/\, m = (m-1)/2 \geqslant \sqrt{3g/2 + 3/4}$.

The latter inequality holds because, one can check that $(m-1)^2 \geqslant (m-3)(m-4) + 9$ for all $m \geqslant 4$. Since

$$(m-3)(m-4) + 9 \geqslant 6\lceil (m-3)(m-4)/6 \rceil + 3 = 6g + 3,$$

we have $(m-1)^2/4 \geqslant (6g+3)/4$, and thus $(m-1)/2 \geqslant \sqrt{3g/2 + 3/4}$ as claimed. ☐

References

[ABY63] Louis Auslander, Thomas A. Brown, and John W.T. Youngs. “The Imbedding of Graphs in Manifolds”. In: *Journal of Mathematics and Mechanics* 12.4 (1963), pp. 629–634. URL: http://www.jstor.org/stable/24900840.

[AGLP89] Baruch Awerbuch et al. “Network decomposition and locality in distributed computation”. In: *30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, May 1989, pp. 364–369. DOI: 10.1109/SFCS.1989.63504.

[AORS18] Saeed Akhoondian Amiri et al. “Distributed Domination on Graph Classes of Bounded Expansion”. In: *30th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. Vienna, Austria: ACM Press, July 2018, pp. 143–151. DOI: 10.1145/3210377.3210383.

[ASS19] Saeed Akhoondian Amiri, Stefan Schmid, and Sebastian Siebertz. “Distributed Dominating Set Approximations beyond Planar Graphs”. In: *ACM Transactions on Algorithms* 15.3 (2019), Article No. 39, pp. 1–18. DOI: 10.1145/3093239.

[BBE+24] Marthe Bonamy et al. “Asymptotic Dimension of Minor-Closed Families and Assouad–Nagata Dimension of Surfaces”. In: *Journal of the European Mathematical Society* 26.10 (2024), pp. 3739–3791. DOI: 10.4171/JEMS/1341.

[BGPW25] Marthe Bonamy et al. “Local Constant Approximation for Dominating Set on Graphs Excluding Large Minors”. In: *44th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. Huatulco, Mexico: ACM Press, June 2025. DOI: 10.1145/3732772.3733531.

[BH25] Ravi B. Boppana and Magnús M. Halldórsson. “Approximating Independent Sets in Constant Distributed Rounds”. In: *32nd International Colloquium on Structural Information & Communication Complexity (SIROCCO)*. Ed. by Ulrich Schmid and Roman Kuznets. Vol. 15671. Lecture Notes in Computer Science. Delphi, Greece: Springer, June 2025, pp. 144–158. DOI: 10.1007/978-3-031-91736-3_9.

[CHS+14] Andrzej Czygrinow et al. “Distributed local approximation of the minimum k-tuple dominating set in planar graphs”. In: *International Conference on Principles of Distributed Systems*. Springer. 2014, pp. 49–59.

[CHS+17] A. Czygrinow et al. “Improved distributed local approximation algorithm for minimum 2-dominating set in planar graphs”. In: *Theoretical Computer Science* 662 (2017), pp. 1–8. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2016.12.001. URL: https://www.sciencedirect.com/science/article/pii/S0304397516307071.

[CHW08] Andrzej Czygrinow, Michał Hańćkowiak, and Wojciech Wawrzyniak. “Fast Distributed Approximations in Planar Graphs”. In: *22nd International Symposium on Distributed Computing (DISC)*. Vol. 5218. Lecture Notes in Computer Science. Springer, Sept. 2008, pp. 78–92. DOI: 10.1007/978-3-540-87779-0_6.

[CHW18] Andrzej Czygrinow, Michał Hańćkowiak, and Wojciech Wawrzyniak. “Distributed Approximation Algorithms for the Minimum Dominating Set in K_h-Minor-Free Graphs”. In: *29th International Symposium on Algorithms and Computation (ISAAC)*. Vol. 123. LIPIcs. Jiaoxi, Yilan, Taiwan, Dec. 2018, 22:1–22:12. DOI: 10.4230/LIPIcs.ISAAC.2018.22.

[CHWW19] Andrzej Czygrinow et al. “Distributed $\mathcal{CONGEST}_{BC}$ constant approximation of MDS in bounded genus graphs”. In: *Theoretical Computer Science* 757 (Jan. 2019), pp. 1–10. DOI: 10.1016/j.tcs.2018.07.008.

[CL21] Corinna Coupette and Christoph Lenzen. “A Breezing Proof of the KMW Bound”. In: *Symposium on Simplicity in Algorithms (SOSA)*. Virtual Conference: ACM-SIAM, Jan. 2021, pp. 184–195. DOI: 10.1137/1.9781611976496.21.

[Feu23] Laurent Feuilloley. *Bibliography of distributed approximation beyond bounded degree*. Tech. rep. 2001.08510v4 [cs.CG]. arXiv, Nov. 2023. DOI: 2001.08510v4.

[FGKS13] Pierre Fraigniaud et al. “What can be decided locally without identifiers?” In: *32nd Annual ACM Symposium on Principles of Distributed Computing (PODC)*. Montréal, Québec, Canada: ACM Press, July 2013, pp. 157–165. DOI: 10.1145/2484239.2484264.

[FKP13] Pierre Fraigniaud, Amos Korman, and David Peleg. “Towards a complexity theory for local distributed computing”. In: *Journal of the ACM* 60.5 (Oct. 2013), Article No. 35, pp. 1–26. DOI: 10.1145/2499228.

[GHS13] Mika Göös, Juho Hirvonen, and Jukka Suomela. “Lower Bounds for Local Approximation”. In: *Journal of the ACM* 60.5 (Oct. 2013), Article No. 39, pp. 1–23. DOI: 10.1145/2528405.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[GKM17] Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. “On the Complexity of Local Distributed Graph Problems”. In: *49th Annual ACM Symposium on Theory of Computing (STOC)*. Montreal, Canada: ACM Press, June 2017, pp. 784–797. DOI: 10.1145/3055399.3055471.

[Gro93] Mikhael Gromov. *Geometric Group Theory: Asymptotic invariants of infinite groups*. Cambridge University Press, 1993.

[HKO+25] Ozan Heydt et al. “Distributed domination on sparse graph classes”. In: *European Journal of Combinatorics* 123 (Jan. 2025), p. 103773. DOI: 10.1016/j.ejc.2023.103773.

[HLS14] Miikka Hilke, Christoph Lenzen, and Jukka Suomela. “Brief Announcement: local approximability of minimum dominating set on planar graphs”. In: *33rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*. Paris, France: ACM Press, July 2014, pp. 344–346. DOI: 10.1145/2611462.2611504.

[KMW16] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. “Local Computation: Lower and Upper Bounds”. In: *Journal of the ACM* 63.2 (Mar. 2016), Article No. 17, pp. 1–44. DOI: 10.1145/2742012.

[KSV21] Simeon Kublenz, Sebastian Siebertz, and Alexandre Vigny. “Constant Round Distributed Domination on Graph Classes with Bounded Expansion”. In: *28th International Colloquium on Structural Information & Communication Complexity (SIROCCO)*. Ed. by Tomasz Jurdziński and Stefan Schmid. Vol. 12810. Lecture Notes in Computer Science. Wrocław, Poland: Springer, Cham, June 2021, pp. 334–351. DOI: 10.1007/978-3-030-79527-6_19.

[KW05] Fabian Kuhn and Roger Wattenhofer. “Constant-time distributed dominating set approximation”. In: *Distributed Computing* 17.4 (2005), pp. 303–310. DOI: 10.1007/s00446-004-0112-5.

[KYK80] Tohru Kikuno, Noriyoshi Yoshida, and Yoshiaki Kakuda. “The NP-Completeness of the Dominating Set Problem in Cubic Planar Graphs”. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E63-E.6 (1980), pp. 443–444.

[Len11] Christoph Lenzen. “Synchronization and symmetry breaking in distributed systems”. PhD Thesis. ETH Zurich, Switzerland, Dec. 2011.

[LOW08] Christoph Lenzen, Yvonne Anne Oswald, and Roger Wattenhofer. “What Can Be Approximated Locally? – Case Study: Dominating Sets in Planar Graphs”. In: *20th Annual ACM Symposium on Parallel Algorithms and Architec-

tures (*SPAA*). Munich, Germany: ACM Press, June 2008, pp. 46–54. DOI: 10.1145/1378533.1378540.

[LPW13] Christoph Lenzen, Yvonne-Anne Pignolet, and Roger Wattenhofer. “Distributed minimum dominating set approximations in restricted families of graphs”. In: *Distributed Computing* 26.2 (Mar. 2013), pp. 119–137. DOI: 10.1007/s00446-013-0186-z.

[LW08] Christoph Lenzen and Roger Wattenhofer. “Leveraging Linial’s Locality Limit”. In: *22nd International Symposium on Distributed Computing* (*DISC*). Vol. 5218. Lecture Notes in Computer Science. Springer, Sept. 2008, pp. 394–407. DOI: 10.1007/978-3-540-87779-0_27.

[MT01] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. The Johns Hopkins University Press, 2001.

[NS95] Moni Naor and Larry Stockmeyer. “What Can Be Computed Locally”. In: *SIAM Journal on Computing* 24.6 (Dec. 1995), pp. 1259–1277. DOI: 10.1137/S0097539793254571.

[RG20] Václav Rozhoň and Mohsen Ghaffari. “Polylogarithmic-Time Deterministic Network Decomposition and Distributed Derandomization”. In: *52nd Annual ACM Symposium on Theory of Computing* (*STOC*). Chicago, IL, USA: ACM Press, June 2020, pp. 350–363. DOI: 10.1145/3357713.3384298.

[RWZ09] Pascal von Rickenbach, Roger Wattenhofer, and Aaron Zollinger. “Algorithmic Models of Interference in Wireless Ad Hoc and Sensor Networks”. In: *IEEE/ACM Transactions on Networking* 17.1 (Feb. 2009), pp. 172–185. DOI: 10.1109/TNET.2008.926506.

[Suo13] Jukka Suomela. “Survey of Local Algorithms”. In: *ACM Computing Surveys* 45.2 (Feb. 2013), Article No. 24, pp. 1–40. DOI: 10.1145/2431211.2431223.

[WAF02] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks”. In: *21st Annual Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM*). Vol. 3. June 2002, pp. 1597–1604. DOI: 10.1109/INFCOM.2002.1019411.

[WY07] Peng-jun Wan and Chih-wei Yi. “On the Longest Edge of Gabriel Graphs in Wireless Ad Hoc Networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 18.1 (Jan. 2007), pp. 111–125. DOI: 10.1109/TPDS.2007.253285.

Chapter IV

MDS in $K_{2,t}$-minor-free graphs

Abstract

We show that graphs excluding $K_{2,t}$ as a minor admit an $f(t)$-round 50-approximation deterministic distributed algorithm for Minimum Dominating Set. The result extends to Minimum Vertex Cover. Though fast and approximate distributed algorithms for such problems were already known for H-minor-free graphs, all of them have an approximation ratio depending on the size of H. To the best of our knowledge, this is the first example of a large non-trivial excluded minor leading to fast and constant-approximation distributed algorithms, where the ratio is independent of the size of H. A new key ingredient in the analysis of these distributed algorithms is the use of *asymptotic dimension*.

Acknowledgements

This chapter is a version of a work with Marthe Bonamy, Cyril Gavoille and Alexandra Wesolek published in PODC'25 [BGPW25b] adapted to the thesis. The authors would like to thank Linda Cook and Sergey Norin for inspiring discussions, and the reviewers for helpful comments.

Contents

IV.1 Introduction

As seen in Chapter III, MINIMUM DOMINATING SET (MDS) (and its weaker version, MINIMUM VERTEX COVER (MVC)) is a famous minimization problem on graphs, known to be NP-complete even in cubic planar graphs [GJ79, KYK80]. The goal is to find a smallest subset of vertices that intersects all radius-1 balls (MDS) or all edges (MVC). The distributed version of this problem is important, and an extended history of MDS in the LOCAL model can be found in Section III.1.

Constant-Round Algorithms. Because of the lower bound of [KMW16] in general graphs, achieving constant ratio approximation in a constant number of rounds is not possible. More precisely, every constant-approximation LOCAL algorithm requires $\Omega(\sqrt{\log n/\log\log n})$ rounds, and this holds for MDS and MVC. Therefore, we need to focus on restricted graph classes to obtain such results.

The literature is abundant in this direction. For instance, on regular graphs, i.e., graphs where all vertices have the same degree, we achieve a 2-approximation for MVC in 0 rounds (take all vertices[1]). Similarly, a 6-approximation in unit-disk graphs can be achieved by taking all the vertices incident to an edge – See the excellent survey of [Suo13] and the references therein. For the more difficult MDS problem, distributed algorithms have been developed for various classes of graphs, including (but not limited to): outerplanar graphs [BCGW21], planar graphs [LPW13, HLS14, HKO+25], bounded-genus graphs [ASS19, BGPW25a], graphs excluding topological minors [CHW18], graphs with sublogarithmic expansion [ASS19] or with bounded expansion [KSV21, LW24, HKO+25]. For instance, the approximation ratio for MDS in planar graphs has been improved from 130 [LPW13] to 52 [Waw14], and recently down to $11+\varepsilon$ [HKO+25].

[1] This holds by observing that such a graph contains $kn/2$ edges where k is the degree of each vertex, while a set on p vertices intersects at most pk edges.

H-minor-free With Large H. Most of the graph classes cited above can be expressed as H-minor-free graphs for some specific minor H (cf. Section IV.1), but the results for graphs with bounded expansion are more general. More precisely, [KSV21] presented a constant-round LOCAL algorithm with approximation ratio $\nabla_1(G)^{O(t\nabla_1(G))}$ if G excludes $K_{t,t}$ as subgraph, where $\nabla_r(G)$ is the maximal edge density of a depth-r minor of G and $t = O(\nabla_1(G))$. [HKO+25] have improved the approximation ratio to $\nabla_0(G)\cdot\nabla_1(G)^{O(s\nabla_1(G))}$ if G excludes $K_{s,t}$ as subgraph, at the cost of a larger $O_t(1)$-round complexity. If G excludes $K_{3,t}$ as subgraph, the approximation ratio improves to $(2+\varepsilon)\cdot(2\nabla_1(G)+1)$ for every $\varepsilon > 0$, where the round complexity is $O_{\varepsilon,t}(1)$. For graphs of girth at least 7, it is possible to achieve a better approximation ratio of $O(\sqrt{\nabla_1(G)})$ in 3 rounds [LW24].

Obviously, if G excludes H as minor, it excludes H as a depth-r minor. As a consequence, $\nabla_r(G) \leqslant \delta(H)$, where $\delta(H)$ is the maximum edge density of a graph excluding H as minor. It is well-known [Kos84, Tho01] that $\delta(K_t) = \Theta(t\sqrt{\log t})$, and more generally that $\delta(H) = \Theta(t\sqrt{\log d})$ [RW16], where $t = |V(H)|$ and $d = |E(H)|/t < \mathrm{pw}(H) + 1$. So, $\delta(K_{s,t}) = \Theta(t\sqrt{\log s})$. More specifically, we have $\delta(K_{3,t}) \leqslant (t+3)/2$ [KP10]. Therefore, for graphs excluding K_t as minor, the result of [KSV21] implies an approximation ratio of $t^{O(t^2\sqrt{\log t})}$ with $O(1)$-round complexity. For $K_{s,t}$-minor-free, [HKO+25] implies an approximation ratio of $t^{O(st\sqrt{\log s})}$ with $O_t(1)$-round complexity. For $K_{3,t}$-minor free graphs, the approximation ratio becomes $(2+\varepsilon)\cdot(t+4)$. See Section IV.1 for a compilation of best known results for various H.

Our Contributions. In this chapter, we concentrate our attention on H-minor-free graphs when H has many vertices, that is, t vertices for some arbitrarily large parameter $t \in \mathbb{N}$. To the best of our knowledge (see Section IV.1), no $f(t)$-round and constant-approximation LOCAL algorithm for MDS in H-minor-free graphs is known, excepted perhaps for the trivial case where H is a subgraph of a path with t vertices. Indeed, in this case the graph G has diameter at most $t-1$, and thus an MDS can be solved exactly in $t-1$ rounds[2].

For MDS, we show that:

- $K_{2,t}$-minor-free graphs have a $O_t(1)$-round 50-approximation LOCAL algorithm.
- These graphs also have an $O(1)$-round $(2t+1)$-approximation LOCAL algorithm.

All the algorithms are deterministic, since in general constant-round randomized LOCAL algorithms are not possible if high probability guarantee is required.

MINOR-FREE GRAPHS	APPROX. RATIO	#ROUNDS	REFERENCES
trees (K_3)	3	2	Folklore [3]
outerplanar (K_4, $K_{2,3}$)	5	2	[BCGW21]
planar (K_5, $K_{3,3}$)	$11+\varepsilon$	$O_\varepsilon(1)$	[HKO+25]
$K_{1,t}$-minor-free	t	0	Folklore [4]
$K_{2,t}$-minor-free	$2t-1$	3	**Theorem IV.3.4**
$K_{2,t}$-minor-free	**50**	$O_t(1)$	**Theorem IV.3.1**
$K_{s,t}$-minor-free	$t^{O(st\sqrt{\log s})}$	$O_{\varepsilon,t}(1)$	[HKO+25]
K_t-minor-free	$t^{O(t^2\sqrt{\log t})}$	7	[KSV21]

Table IV.1: Constant-round approximation distributed algorithms for Minimum Dominating Set on H-minor-free graphs, for various H. The bottom part of the table is about large H, on t vertices where t may be arbitrarily large.

Our approach toward Theorem IV.3.1 is to design an algorithm as simple as possible and push all the complexity to its analysis, following a long tradition [Waw14, BCGW21]. The intuition here is that a highly-connected $K_{2,t}$-minor-free graph has bounded radius, so the difficulty lies in handling small cuts, especially since we cannot decide locally if a vertex belongs to a small cut. We treat all vertices that are in a cut of size 1

[2] In the LOCAL model, after D rounds of communication, each vertex u of a diameter-D graph knows entirely G and its identifier in G. After this communication step, u can therefore compute an optimal dominating set in a consistent way with centralized brute-force and deterministic algorithm.

[3] If there are at least three vertices, take all vertices with degree at least two, cf. [LPW13, BCGW21]. This requires two rounds from the model, because the vertices do not know their degree and need one round to count their neighbors (by counting the number of received messages).

[4] Take all the vertices. Such graphs have degree at most $t-1$, thus this is a 0-round t-approximation since every dominating set has size at least $n/(\Delta+1)$ where Δ is the maximum degree of the graph.

or 2 in their small-distance neighborhood as we would vertices that are in a cut of size 1 or 2 in the whole graph (take all vertices in a cut of size 1, take all vertices in a cut of size 2 except those which are clearly a bad idea), then argue what remains[5] is a number of connected components of bounded weak radius, which we can thus solve optimally by brute-force. Though the latter part comes with its own interesting challenges which we thankfully mostly outsource[6] to a paper of Ding [Din17], the major conceptual contribution is in the analysis of the first part. To argue that taking all vertices that are locally separating (and similarly for vertices in a local cut of size 2) is not too costly, we need to give some global discharging argument. We were once again able to do this using recent results on the asymptotic dimension.

[5] While 3-connected $K_{2,t}$-minor-free graphs may have unbounded radius, such graphs admit many local 1- or 2-cuts, see Lemma IV.3.2.

[6] While the paper is only available as a preprint and has seemingly not gone through a reviewing process, it seems to be generally considered to be correct and has even been refined in a doctoral thesis [Sol19]. For our own peace of mind, we have triple-checked that the pieces we need from that paper really do hold.

Sketch of the Main Algorithm (Theorem IV.3.1). The algorithm computes an approximation of Minimum Dominating Set in a $K_{2,t}$-minor-free graph G. It has three main steps:

1. Compute the set X of all vertices in “local” 1-cuts, and add them to the solution.
2. Compute the set I of all “interesting” vertices in “local” 2-cuts, and add them to the solution.
3. Compute an optimal dominating set of all other undominated vertices in each component of $G - (X \cup I)$ using a brute-force approach and add them to the solution.

Intuitively, a “local” k-cut is a minimal set of vertices that locally (up to some bounded radius) looks like a standard k-cut. This radius is a function of the size of H and of $k \in \{1, 2\}$. And, a vertex u in a “local” 2-cut $\{u, v\}$ is “interesting” if v does not dominate all vertices, except for at most one component attached to $\{u, v\}$. This is a rough explanation, and all formal definitions of “local” k-cuts and “interesting” vertices can be found in Sections IV.2 and IV.2.1.

The main challenge is to accurately tune the above radii in Step 1 and 2 to show that the approximation ratio (namely 50) does not depend on the size of H, but only on the asymptotic dimension of the class and its control function. In contrast, the round complexity essentially relies on the diameter of the components as defined in Step 3, which is a function of the radii defined above.

IV.2 Preliminaries

General Definitions. A graph without true twins is a graph such that no two distinct vertices u and v are true twins, i.e. are such that $N[u] = N[v]$. The *true-twin-less* graph associated to G is the largest subgraph of G with no true twins. Notice that there is a unique such unlabelled graph G^- and that it can be computed in a constant number of rounds in the LOCAL model. Furthermore, $\text{MDS}(G^-) = \text{MDS}(G)$.

Local Connectivity. The aim of this definition is to study cuts that can be recognized using a LOCAL algorithm. Recall that a k-cut of a graph G is a minimal subset of k vertices whose removal increases the number of connected components of G. E.g., 1-cuts are a.k.a. cut-vertices. On classes of bounded asymptotic dimension, the set of local k-cuts is well-behaved for $k \leqslant 2$ (see Lemma IV.2.2).

Here is a formal definition of a local cut[7]. By $N^r[v]$, we denote the set of all vertices at distance at most r of v in G. A subset of vertices C of a graph G is an *r-local k-cut* if all vertices of C are pairwise at distance at most r in G, and C is a k-cut of $G\left[\bigcup_{v\in C} N^r[v]\right]$. We say G is r-locally k-connected if G has no r-local k-cuts. If there are no r-local k-cuts, then there are no r'-local k-cuts for any $r' > r$. Note that a k-cut is a $|V(G)|$-local k-cut. Therefore, a locally k-connected graph is k-connected, that is, local connectivity is a stronger notion than connectivity. All cuts that will be considered from now on are minimal cuts, i.e., no proper subset of the cut is also a cut with the "same" connected components. Intuition is not always an ally when it comes to r-local k-cuts, however we use the notion in a fairly basic manner here.

For a definition of asymptotic dimension (and some intuition), see Section III.2.

[7] We did not find this exact same notion anywhere else in the literature, though it was probably considered previously.

IV.2.1 *Bounding the Number of Local* 1*-cuts and* 2*-cuts*

We first bound the number of vertices in local 1-cuts and the number of vertices in so-called interesting local 2-cuts.

> **Lemma IV.2.1.** *For any $d \in \mathbb{N}$, and any graph class $\mathscr{C}$ of asymptotic dimension at most d, there exists $c_{IV.2.1}(d)$ and $m_{IV.2.1}(\mathscr{C})$ such that for all graphs $G \in \mathscr{C}$, the number of $m_{IV.2.1}(\mathscr{C})$-local 1-cuts in G is at most $c_{IV.2.1}(d)\,\mathrm{MDS}(G)$.*

The proof of this can be found in Section IV.4.1.

To extend Lemma IV.2.1 to 2-cuts, we need some restriction on the 2-cuts considered: for example, a large tree with a single vertex adjacent to all its vertices admits many 2-cuts but has a dominating set of size 1. This motivates the following definition, which we will only use for $r \geqslant 2$.

A vertex v is *r-interesting* if there exists some r-local 2-cut $c = \{u, v\}$ such that:

- $N[v] \not\subseteq N[u]$ and
- at least two connected components of $G[N^r[c]] - c$ contain each a vertex non-adjacent to u.

We are now ready to state the corresponding lemma for 2-cuts.

> **Lemma IV.2.2.** *For any $d \in \mathbb{N}$, and any graph class $\mathscr{C}$ of asymptotic dimension at most d, there exists $c_{IV.2.2}(d)$ and $m_{IV.2.2}(\mathscr{C})$ such that for*

all graphs $G \in \mathscr{C}$, the number of interesting vertices in $m_{IV.2.2}(\mathscr{C})$-local 2-cuts of G is at most $c_{IV.2.2}(d) \cdot \mathrm{MDS}(G)$.

The proof of this can be found in Section IV.4.2.

IV.3 Constant Approximation for Minimum Dominating Set

Intuition and Explanation. One can assume that the graph contains no true twins, just like in the algorithm of Theorem IV.3.4. The main idea of our algorithm is to take all vertices in 1-cuts and 2-cuts, in order to reduce the problem to 3-connected graphs, where we can solve Minimum Dominating Set in constant time $O(t)$. However, it is not possible to do this in constant round-complexity in the LOCAL model. Therefore, instead of considering k-cuts we consider sets of k vertices that resemble a k-cut locally. With a little luck, those vertices are actually 1-cuts, but not all local 1-cuts are 1-cuts. Indeed, consider a very long cycle. All vertices are local 1-cuts but none are global 1-cuts. However, we can show that, if the graph has bounded asymptotic dimension, there exists some constant r (that does not depend on the graph) such the number of r-local 1-cuts is bounded above by a function linear in $\mathrm{MDS}(G)$. Therefore, our algorithm can take all local 1-cuts in the returned set. The case of local 2-cuts is more complicated: there are graphs with $\omega(\mathrm{MDS}(G))$ many vertices in 2-cuts. Indeed, consider a clique G of size n. Take an arbitrary vertex of the clique u, and for all vertices $v \neq u$ of the clique, add a new vertex x_{uv} attached to $\{u, v\} = N(x_{uv})$. This creates a graph G which can be dominated solely by the vertex u. However, all vertices of the original clique are in some minimal 2-cut, as $\{u, v\}$ separates x_{uv} from the rest of the clique – there is an unbounded number of vertices in minimal 2-cuts. This leads us to the definition of interesting vertices in 2-cuts, which we mentioned in Section IV.2.1 and recall here. A vertex $v \in C$ is *r-interesting* for some $r \geqslant 2$ if there exists some r-local 2-cut $c = \{u, v\}$ such that:

- $N[v] \not\subseteq N[u]$ and
- at least two connected components of $G[N^r[c]] - c$ contain each a vertex non-adjacent to u.

The first condition is intuitive: one better take u instead of v if $N[v] \subseteq N[u]$. The rough idea behind the second condition is that it allows us to create a nice mapping from interesting vertices to a minimum dominating set. In more detail, we give a tree-like structure to the vertices in 2-cuts, and show the second condition gives us the existence of some vertex d in an MDS satisfying the following property: d is a successor of u in the tree-like structure, and is at bounded distance from

u in the tree-like structure. Now, every interesting vertex u can charge this vertex d. We then show that this d does not receive too many charges, because of the tree-like structure of the interesting 2-cuts, and this allows us to bound the number of interesting vertices. Let us do a quick recap of what the algorithm has done until now: it has taken all local minimal 1-cuts and all interesting vertices in local minimal 2-cuts. Let us consider an arbitrary local minimal 2-cut $\{u, v\}$. There are three cases. First, if both u and v are interesting, the algorithm has taken both vertices in its return set. All components attached to $\{u, v\}$ can now be solved independently. Secondly, if u is interesting and v is not interesting, the algorithm has taken u in its return set. Either $N[v] \subseteq N[u]$ and all components attached to $\{u, v\}$ can now be solved independently, or u dominates all but one component attached to $\{u, v\}$. In any case, all components attached to $\{u, v\}$ can now be solved independently too, as only one undominated component exists. Finally, if neither u nor v is interesting, then as the graph is without true twins, one of u or v dominates all but one component attached to $\{u, v\}$. Therefore, all components attached to $\{u, v\}$ can now be solved independently too. Now, one can prove that the undominated vertices form connected components of bounded diameter, and our algorithm can brute-force and G-dominate the rest of the vertices in constant time.

The Algorithm. Let $t \geqslant 2$ be an integer. The following algorithm computes an approximation of Minimum Dominating Set on the class $\mathscr{C}_t$ of $K_{2,t}$-minor-free graphs. The algorithm is divided into four steps:

1. Remove true twins from the graph.
2. Compute the set X of all vertices in minimal $m_{IV.2.1}(\mathscr{C}_t)$-local 1-cuts, and add them to the returned Dominating Set.
3. Compute the set I of minimal $m_{IV.2.2}(\mathscr{C}_t)$-interesting vertices which are in $m_{IV.2.2}(\mathscr{C}_t)$-local 2-cuts, and add them to the returned Dominating Set.
4. Let U be the set of already dominated vertices that have no undominated neighbors. Dominate all other undominated vertices, in every component of $G - (X \cup I \cup U)$, using a brute-force approach.

A more formal description of the algorithm is given below:

Algorithm 5: Constant approximation for Minimum Dominating Set

Require: An integer t, and G a $K_{2,t}$-minor-free graph
Ensure: S is a dominating set of G with $|S| = O(\mathrm{MDS}(G))$
$G \leftarrow$ true-twin-less graph associated to G
$S \leftarrow \{v \in V(G) \mid \{v\}$ is a $m_{IV.2.1}(\mathscr{C}_t)$-local minimal 1-cut of $G\}$
$S \leftarrow S \cup \{v \in C \mid v$ is a $m_{IV.2.2}(\mathscr{C}_t)$-interesting vertex of a $m_{IV.2.2}(\mathscr{C}_t)$-local minimal 2-cut of $G\}$
$S \leftarrow S \cup$ (brute-forced minimum set of G that dominates $G - N[S]$)

We can now state the main theorem of this section.

Theorem IV.3.1. *For every integer $t \geqslant 2$, Algorithm 5 is an $O_t(1)$-round 50-approximate deterministic LOCAL algorithm for Minimum Dominating Set on $K_{2,t}$-minor-free graphs.*

Proof. Algorithm 5 clearly outputs a dominating set of G. Thus, by Lemmas IV.2.1 and IV.2.2, the approximation ratio of the algorithm is $c_{IV.2.1}(1) + c_{IV.2.2}(1) + 1 = 50$. □

It remains to argue that the number of rounds is bounded. To do this, we need to argue that the brute-forcing performed takes constant time.

Lemma IV.3.2. *For every integer t, if $\mathscr{C}_t$ is the class of $K_{2,t}$-minor-free graphs, there exists $m_{IV.3.2}(t)$ such that for every $G \in \mathscr{C}_t$, if X is the set of vertices in $m_{IV.2.1}(\mathscr{C}_t)$-local 1-cuts, I is the set of $m_{IV.2.2}(\mathscr{C}_t)$-locally interesting vertices of G, and $U = \{u \in N[I \cup X] \mid N[u] \subseteq N[I \cup X]\}$, then every connected component of $G \setminus (I \cup X \cup U)$ has diameter at most $m_{IV.3.2}(t)$.*

The proof of this can be found in Section IV.4.3.

The asymptotic dimension of $K_{2,t}$-minor-free graphs is 1 by [BBE+24] – $K_{2,t}$ is planar so $K_{2,t}$-minor-free graphs have bounded treewidth by the grid minor theorem. For $\mathscr{C}_t$ the class of $K_{2,t}$-minor-free graphs, the running time is

$$\begin{aligned}&\max\{m_{IV.2.1}(\mathscr{C}_t), m_{IV.2.2}(\mathscr{C}_t), m_{IV.3.2}(t)\}\\&= 3\max\{f(5)+2, f(11)+5\} + g(t) + 3,\end{aligned}$$

where f is the control function of the class of $K_{2,t}$-minor-free graphs, and g the linear function given in [Din17, Lemma 6.3]. Choosing $f(r) = (5r+18)t$ suffices, see [BBE+24, Lemma 7.1]; and the running time of the algorithm is linear in t.

The observant reader may be struck by the fact that the roles of t and of the asymptotic dimension seem disjoint. This can be highlighted with the following variant, which computes a $(c_{IV.2.1}(d) + c_{IV.2.2}(d) + 1)$-approximation of MDS in a class of asymptotic dimension d, given its control function, with running time that depends on f and the largest $K_{2,t}$-minor of the input graph but does not require prior knowledge of it.

Algorithm 6: Constant approximation for Minimum Dominating Set in a bounded asdim class

Require: An integer d, a control function f, and G a graph in a class $\mathcal{G}$ of asymptotic dimension d with control function f
Ensure: S is a dominating set of G with $|S| = O(\mathrm{MDS}(G))$
$G \leftarrow$ true-twin-less graph associated to G
$S \leftarrow \{v \in V(G) \mid \{v\}$ is a $m_{IV.2.1}(\mathcal{G})$-local minimal 1-cut of $G\}$
$S \leftarrow S \cup \{v \in C \mid v$ is a $m_{IV.2.2}(\mathcal{G})$-interesting vertex of a $m_{IV.2.2}(\mathcal{G})$-local minimal 2-cut of $G\}$
$S \leftarrow S \cup$ (brute-forced minimum set of G that dominates $G - N[S]$)

We can now state the following stronger version of Theorem IV.3.1.

Theorem IV.3.3. *For every integer d and control function f, Algorithm 6 is an $O_t(1)$-round $(c_{IV.2.1}(1) + c_{IV.2.2}(1) + 1)$-approximate deterministic LOCAL algorithm for* Minimum Dominating Set *on graphs in a class of asymptotic dimension d with control function f, where t is the (unknown) size of a largest $K_{2,t}$-minor in the input graph.*

As an added note, if one wishes to have an algorithm for Minimum Vertex Cover instead of Minimum Dominating Set, it suffices to take all $m_{IV.2.2}(\mathscr{C}_t)$-local 2-cuts instead of just $m_{IV.2.2}(\mathscr{C}_t)$-interesting vertices. On the analysis side, one can prove a simpler variant of Lemma IV.2.2 that bounds the number of vertices in local 2-cuts with respect to the size of a minimum vertex cover. Therefore, both Theorem IV.3.1 and Theorem IV.3.3 extend to the context of Minimum Vertex Cover.

We conclude the "non-technical" part of the chapter with the following result, which shows a different trade-off: linear approximation in constant number of rounds.

Theorem IV.3.4. *For every integer $t \geqslant 2$, there is a 3-round $(2t-1)$-approximate deterministic LOCAL algorithm (resp. t-approximate) for* Minimum Dominating Set *(resp.* Minimum Vertex Cover*) on $K_{2,t}$-minor-free graphs.*

As outerplanar graphs are a subfamily of $K_{2,3}$-minor-free graphs, this result generalizes the 5-approximation algorithm of [BCGW21] on outerplanar graphs in the case of the LOCAL model. The proof of Theorem IV.3.4 can be found in Section IV.4.4.

IV.4 Proofs

IV.4.1 Proof of Lemma IV.2.1: Bounding the Number of Vertices in Local 1-cuts

We will need the following lemma for the rest of the proofs in this section.

Lemma IV.4.1. *Let G be a graph and let $R_0, R_1, \ldots, R_k \subseteq V(G)$ subsets of vertices such that all $N[R_i]$ are pairwise disjoint. Then*

$$\sum_{i=0}^{k} \mathrm{MDS}(G, R_i) \leqslant \mathrm{MDS}(G)\,.$$

Let us now prove Lemma IV.2.1.

We did not try to optimize the constants $c_{IV.2.1}(d)$ and $m_{IV.2.1}(\mathscr{C})$. Let f be the d-dimensional control function of the graph class. We prove the lemma for $c_{IV.2.1}(d) = 3 \cdot (d+1)$ and $m_{IV.2.1}(\mathscr{C}) = f(5) + 2$. Without loss of generality, we can assume G is connected. We will first prove there are not too many 1-cuts in some $S \subseteq V(G)$ compared to $\mathrm{MDS}(N[S])$.

> **Claim IV.4.2.** *Let G be a graph and $S \subseteq V(G)$. Let C be the set of minimal 1-cuts of G. Then $|C \cap S| \leqslant 3 \cdot \mathrm{MDS}(G, N[S])$.*

Let us prove this claim. Without loss of generality, we can assume G is connected. Let C be the set of 1-cuts of G and B the set of maximal 2-connected components of G. Let T be the bipartite graph with vertex set $B \cup C$ and with edge set $E(T) = \{(b, c) \in B \times C \mid c \in b\}$. T is sometimes called the block-cut tree of G and can be proven to be a tree. Moreover, note that all leaves of T are in B. Let $S \subseteq V(G)$, and $D \subseteq N^2[S]$ a dominating set of $N[S]$. We prove that $|C \cap S| \leqslant 3|D|$. If $C \cap S = \varnothing$, we are done. Otherwise, root T at an arbitrary cut-vertex r. We have the following:

> **Claim IV.4.3.** *Let $c \in C \cap S$. Then there exists b such that $c \in b \in B$ such that $b \cap D \neq \varnothing$.*

This is because all vertices of $C \cap S$ must be dominated by some vertex of D. Therefore, either $c \in D$ and then we are done as there exists some $b \in B$ such that $c \in b$, or either there exists $a \in D \cap N[c]$. This a must be contained in some neighboring 2-connected component, therefore there exists $b \in B$ such that $a \in b$, i.e. $b \cap D \neq \varnothing$.

In the following, we create a mapping from vertices of $C \setminus D$ to D. Consider $c \in C \setminus D$. There are 3 different cases.

- Either c has a child $b \in B$ with some $d \in b \cap D \cap N(b)$, and in this case, we map c to d.
- Or either c has a descendant $c' \in C \cap N(c)$ (at distance 2 in T), and the previous claim still applies to c': c' has a descendant $b' \in B$ in T such that there exists $d \in d \cap D \cap N(c')$. We map c to d.
- Or c has no descendant in $C \cap \cap N(c)$. In this case, as c cannot be a leaf of T, there exists a child $b \in B$ of c, with the property that $b \setminus C \neq \varnothing$. As D dominates $N[S]$, b must contain a vertex $d \neq c$ of D. We map c to b.

Therefore, we created a mapping from vertices of $C \setminus D$ to vertices of D. Furthermore, each vertex from D can appear at most twice in a preimage. Indeed, for some $d \in D$, only an ancestor in C at distance 1 or 3 in T can be mapped to it. In conclusion, $|C| \leqslant |C \cap D| + |C \setminus D| \leqslant |D| + 2|D| = 3|D|$.

We can now get a bound on the number of local 1-cuts.

> **Claim IV.4.4.** *Let G be a graph of asymptotic dimension d with control function f. Then, the number of $(f(5)+2)$-local 1-cuts is bounded by $3(d+1) \cdot \mathrm{MDS}(G)$.*

This will directly imply the statement of the lemma. Let r a positive integer and C be the set of $(f(5)+2)$-local cuts of G. Fix $B \subseteq V(G)$ such that $G[B]$ has weak diameter $f(5)$. We claim the following:

Claim IV.4.5. *Every $(f(5)+2)$-local 1-cut of G that is in B is also a 1-cut of $G[N^2[B]]$.*

Proof of claim. Indeed, let $v \in B$ be a $(f(5)+2)$-local 1-cut of G and $a, b \in N(v)$ separated by v, i.e. every ab-path of G of $N^{f(5)+2}[v]$ contains v. Notice that $N^2[B] \subseteq N^{f(5)+2}[v]$ because B has weak diameter at most $f(5)$. a and b are separated by v in $G[N^2[B]]$, because $a, b \in N^2[B]$ and because no ab-path in $N[B] \setminus \{v\}$ exists. Therefore, Claim IV.4.5 is proven. ◇

Using this fact and with the help of Claim IV.4.2, we can bound the number of 1-cuts of $G[N^2[B]]$ in B by $3 \cdot \mathrm{MDS}(G[N^2[B]], N[B]) \leqslant 3 \cdot \mathrm{MDS}(G, N[B])$. By the definition of the asymptotic dimension, there is a cover $B_0, B_1, \ldots, B_d$ of G where the 5-components (connected components in G^5) of a B_i have weak diameter at most $f(5)$. Note that each B_i contains distinct 5-components which are of distance at least 6 from each other. With an abuse of notation, when we write $B \in B_i$ we mean that B is a 5-component of B_i. That is, we treat B_i as the set of its 5-components. As $B_0, B_1, \ldots, B_d$ is a cover of $V(G)$ by subsets of diameter at most $f(5)$, we can bound the number of $(f(5)+2)$-local 1-cuts of G by summing the number of 1-cuts of all the $G[N^2[B_i]]$'s. We get

$$|C| \leqslant \sum_{i=0}^{d} \sum_{B \in B_i} 3 \cdot \mathrm{MDS}(G, N[B]) .$$

Notice that because the B_i's are partitioned into their 5-components, all elements of $\{N^2[B] \mid B \text{ is a 5-component of } B_i\}$ are pairwise disjoint.

Therefore, by Lemma IV.4.1, we get

$$|C| \leqslant \sum_{i=0}^{d} 3 \cdot \mathrm{MDS}(G) = 3(d+1) \cdot \mathrm{MDS}(G) .$$

This finishes the proof of Lemma IV.2.1.

IV.4.2 Proof of Lemma IV.2.2: Bounding the Number of Interesting Vertices

When discussing global 2-cuts and not local ones, we say v is interesting if there exists a 2-cut $c = \{u, v\}$ such that:

- $N[v] \not\subseteq N[u]$ and
- at least two connected components of $G - c$ contain each a vertex non-adjacent to u.

Moreover, v is called a *friend* of u, and a 2-cut $\{u, v\}$ where u is interesting and v is a friend of u is called *interesting*. If u only has the second property, it is called *almost-interesting*.

Two 2-cuts c_1, c_2 of G are said to be *crossing* if the two following conditions are satisfied:

- the two vertices of c_1 are in different components of $G - c_2$, and
- the two vertices of c_2 are in different components of $G - c_1$.

Before bounding the number of interesting vertices in local 2-cuts, we first need to arrange the interesting cuts in a tree-like fashion, i.e. we want to build a bounded number of families of 2-cuts such that each member of the family contains interesting 2-cuts that are all pairwise non-crossing, and such that each interesting vertex appears in one family member, along with one of its friends.

One can easily see that a family of size 2 does not suffice by considering C_6. If we want to only take interesting cuts in this graph, we need to take the 3 opposing cuts. In more detail, if the vertices $\{a, b, c, d, e, f\}$ of C_6 appear in clockwise order a, b, c, d, e and f, then we need to take the interesting cuts $\{a, d\}$, $\{b, e\}$ and $\{c, f\}$.

To create our new 2-cut forest for interesting vertices, we need to introduce SPQR trees.

SPQR Trees. An SPQR tree is a tree data structure that represents the decomposition of a 2-connected graph into its 3-connected components. The construction of an SPQR tree can be accomplished in linear time and SPQR are known to have applications in dynamic graph algorithms and graph drawing.

An SPQR tree T is an unrooted tree where each node μ corresponds to an undirected skeleton graph G_μ that can be one of the following four types.

- ***S*-node:** G_μ is a cycle containing three or more vertices. This represents series composition in series-parallel graphs.
- ***P*-node:** G_μ corresponds to a dipole graph, a multigraph with two vertices and three or more edges, analogous to parallel composition.
- ***Q*-node:** G_μ corresponds to a dipole connected by two parallel edges: one real and one virtual. This serves as a trivial case for graphs with two parallel edges. We will not consider these types of nodes.
- ***R*-node:** G_μ is a 3-connected graph that is neither a cycle nor a dipole.

Edges xy between nodes in the SPQR tree are associated with two directed virtual edges, one from G_x and the other from G_y. Each edge in G_x can be a virtual edge for at most one edge in the SPQR tree.

The SPQR tree represents a 2-connected graph G_T, constructed as follows. Start from the disjoint union of the graphs G_x. If $xy \in E(T)$ is associated with the virtual edge $ab \in E(G_x)$, and with the virtual edge $cd \in E(G_y)$, then identify a with c and b with d, and delete the two virtual edges. Notably, no two adjacent S or P nodes are allowed, ensuring the uniqueness of the SPQR tree representation for a graph G.

When such conditions are met, the graphs G_x associated with the nodes of the SPQR tree are the triconnected components of G.

Proposition IV.4.6 (folklore)**.** *Let T be an SPQR tree of a graph G (without Q nodes) and let $\{u, v\}$ be a 2-cut of G. Then one of the following holds:*

- *u, v are two endpoints of a virtual edge of an R-node, or*
- *u, v are the two vertices of a P-node that has at least two virtual edges, or*
- *u, v are two endpoints of a virtual edge of a C-node, or*
- *u, v are two non-adjacent vertices of a C-node.*

We can now build our tree-like structure for interesting vertices.

Interesting 2-cuts Forests. An *interesting 2-cut forest* $F = (T_1, T_2, T_3)$ of G consists of three trees T_1, T_2 and T_3 whose vertices contain subsets of $V(G)$. Like SPQR trees, T_i contains nodes that are induced subgraphs with some virtual edges added. Every T_i has nodes can be of three types: A-nodes, for 1-cuts, C-nodes, for interesting 2-cuts, and R-nodes for the rest. T_2 has still C-nodes and R-nodes.

If G has no 1-cut or interesting 2-cut, T_i consists of a single R-node $\mu = G$.

If G has a 1-cut v, we construct T_i inductively. First, we add an A-node μ to T_i. The graph associated to μ consists of the vertex v. Secondly, let $C_1, C_2, \ldots, C_k$ be the connected components of $G - v$. Let G_j be the graph $G[C_j \cup \{v\}]$. Build a corresponding 2-cut tree T_{G_j} for the graph G_j. Let μ_j be the (unique) node in T_{G_j} that contains v. We can now construct T_i by taking the union of all T_{G_j}'s and connecting all μ_j's to μ.

Now, let us handle the case of interesting 2-cuts. We do this by going through an SPQR tree T of G and building sets of 2-cuts P_1, P_2 and P_3 with the following properties:

- for every globally almost-interesting vertex u of G, there exist some i and some friend v of u such that $\{u, v\} \in P_i$, and
- for every i, the 2-cuts in P_i are pairwise non-crossing.

The second property allows us to transform P_i into a 2-cut tree T_i, as follows. First, take some arbitrary $c \in P_i$ and a C-node μ to T_i. The graph associated to μ consists of vertices of $c = \{u, v\}$ and a real (respectively virtual) edge uv if uv is a real (resp. virtual) edge of G. Secondly, let $C_1, C_2, \ldots, C_k$ be the connected components of $G - c$. Let G_j be the graph $G[C_j \cup c]$ to which we add a real edge uv if $uv \notin G[C_j \cup c]$. Build a corresponding interesting 2-cut tree T_{G_j} for the graph G_j. Let μ_j be the (unique) node in T_{G_j} where uv is real. We can now construct T_i by taking the union of all T_{G_j}'s, making uv virtual in all μ_j's, and connecting all μ_j's to μ.

We first build the sets P_1, P_2 and P_3. Then, we prove the two wanted properties in Proposition IV.4.7.

Add the vertices u and v to P_1 if:

- u, v are two endpoints of a virtual edge of an R-node of T, or if
- u, v are the two vertices of a P-node of T that has at least two virtual edges.

Let us now handle the case of C-nodes. Let μ be a C node of the T. If μ contains more than 6 nodes. Let $v_0, v_1, \ldots, v_{k-1}$ be the nodes of μ in the order of the cycle. First, put all $\{u, v\}$ in P_1 if uv is a virtual edge. Secondly, we add some 2-cuts to the P_i's depending on the values of k:

1. If $k \geqslant 8$ and k is even: add to P_1 the 2-cuts $\{v_0, v_{k-3}\}$, $\{v_1, v_{k-4}\}$, $\ldots$, and $\{v_{(k/2)-3}, v_{k/2}\}$, and to P_2 the 2-cuts $\{v_{(k/2)-2}, v_{k-1}\}$ and $\{v_{(k/2)-1}, v_{k-2}\}$.
2. If $k \geqslant 8$ and k is odd: add to P_1 the 2-cuts $\{v_0, v_{k-3}\}$, $\{v_1, v_{k-4}\}$, $\ldots$, $\{v_{((k-1)/2)-3}, v_{(k+1)/2}\}$ and $\{v_{((k-1)/2)-3}, v_{(k-1)/2}\}$. Add to P_2 the 2-cuts $\{v_{((k-1)/2)-2}, v_{k-1}\}$ and $\{v_{((k-1)/2)-1}, v_{k-2}\}$.
3. If $k = 7$, add to P_1 the 2-cut $\{v_0, v_3\}$ and $\{v_0, v_4\}$, to P_2 the 2-cut $\{v_1, v_5\}$ and to P_3 the 2-cut $\{v_2, v_6\}$.
4. If $k = 6$, add to P_1 the 2-cut $\{v_0, v_3\}$, to P_2 the 2-cut $\{v_1, v_4\}$ and to P_3 the 2-cut $\{v_2, v_5\}$.
5. If $k \leqslant 5$ but $G \neq C_k$, suppose without loss of generality that the edge v_0v_1 is virtual. Moreover, suppose that it is the only virtual edge of the C-node. If $k = 5$, add to P_1 the 2-cut $\{v_0, v_2\}$ and to P_2 the 2-cut $\{v_1, v_4\}$.
6. If $k \leqslant 5$ but $G \neq C_k$ and the edges v_0v_1 and v_0v_{k-1} are virtual: add to P_1 all the 2-cuts $\{v_0, v_i\}$ for $i = 2, 3, \ldots, k-2$. Moreover, if $k = 5$, add to P_2 the 2-cut $\{v_1, v_{k-1}\}$.
7. If $k \leqslant 5$ but $G \neq C_k$ and there exists $i \in \{2, 3, \ldots, k-2\}$ such that the edges v_0v_1 and v_iv_{i+1} are virtual: add to P_1 all the 2-cuts $\{v_0, v_j\}$ for $j = 2, 3, \ldots, i$, and add to P_2 all the 2-cuts $\{v_1, v_j\}$ for $j = i+1, i+2, \ldots, k-1$.

Proposition IV.4.7. *Let P_1, P_2 and P_3 be built as described above. Then the two following properties are satisfied:*

- *for every globally interesting vertex u of G, there exist some i and some friend v of u such that $\{u, v\} \in P_i$, and*
- *for every i, the 2-cuts in P_i are pairwise non-crossing.*

Proof. One can easily see that 2-cuts of P_i in the same C-node are taken in such a way that they do not cross. This also applies 2-cuts inside the same nodes inside the same P or R-node. Moreover, 2-cuts inside the different nodes of the SPQR tree cannot cross by Proposition IV.4.6. Therefore, the second property is proven.

We will show that every globally interesting vertex u of G appears in some P_i with one of its friends. As we take all 2-cuts in R-nodes and P-nodes, then by Proposition IV.4.6 the only case where we could have not taken an interesting vertex with its friend is inside a C-node. We prove that is however not the case. Let us consider a C-node μ with k vertices. We go through all the possible cases.

- If $k \geqslant 8$ and k is even: one can verify that all 2-cuts chosen are interesting, and that every vertex of the cycle is in one of the chosen cycles. Therefore, we are done with this case.

- The same applies if $k \geqslant 8$ and k is odd, or if $k = 7, 6$, by checking each case.

- If $G = C_k$ with $k \leqslant 5$, there are no interesting vertices.

- If $k \leqslant 5$ but $G \neq C_k$, and the only virtual edge of μ is v_0v_1. Consider the node connected to μ by this virtual edge. If it is not a P-node, name this node μ'. If it is a P-node, let μ' be one of its neighbor different from μ (it exists as every P-node has degree at least 2 in T). Let us handle the case $k = 5$ first. Notice that by definition, $V(\mu') \setminus V(\mu) \neq \varnothing$. Let us prove that 2-cuts $\{v_0, v_2\}$ and $\{v_1, v_4\}$ are interesting. First, their one component in μ is not fully dominated by either of the vertices of the cuts. Moreover, in each of the two 2-cuts, one of the vertices is not connected to any vertex of $V(\mu') \setminus V(\mu)$. Finally, $N[v_1] \not\subseteq N[v_4]$, $N[v_0] \not\subseteq N[v_2]$, and inversely. Therefore, the 2-cuts $\{v_0, v_2\}$ and $\{v_1, v_4\}$ are indeed interesting. The only vertex that needs checking is now v_3. We claim that v_3 is not interesting. Indeed, any 2-cut in μ containing v_3 is not interesting. Moreover, v_3 cannot have a friend outside of $V(\mu)$ by Proposition IV.4.6. Therefore, v_3 is not interesting. All interesting vertices are now taken in some P_i, and all chosen 2-cuts in μ are chosen. We are done with this case. If $k = 4$ (resp. $k = 3$) then for similar reasons, vertices v_2 and v_3 (resp. v_2) are not interesting. In both cases, all possibly interesting vertices are taken in the 2-cut $\{v_0, v_1\}$ (taken because the edge v_0v_1 is virtual). All other 2-cuts in μ containing either v_0 or v_1 are not interesting. Therefore, we are done with this case.

- If $k \leqslant 5$ but $G \neq C_k$, and the edges v_0v_1 and v_0v_{k-1} are virtual. Consider the node connected to μ by the virtual edge v_0v_1. If it is not a P-node, name this node μ'_1. If it is a P-node, let μ'_1 be one of its neighbor different from μ (it exists as every P-node has degree at least 2 in T). Define similarly the node μ'_2 for the virtual edge v_0v_{k-1}. Let $v'_i \in V(\mu'_i) \setminus V(\mu)$. For $i = 2, 3, \dots, k-2$, the 2-cuts $\{v_0, v_i\}$ are interesting because v_i is not adjacent to v'_1 nor v'_2, and $N[v_0] \not\subseteq N[v_i]$, and inversely. If $k = 5$, the 2-cut $\{v_1, v_{k-1}\}$ is interesting because $v_1v'_2 \notin E(G)$, $v_1v_3 \notin E(G)$ and $N[v_{k-1}] \not\subseteq N[v_1]$, and

$N[v_1] \not\subseteq N[v_{k-1}]$. If $k = 5$, all vertices are taken in interesting 2-cuts. If $k = 4$, v_0 and v_2 are taken in interesting 2-cuts. All other 2-cuts contained only in μ containing either v_0 or v_1 are not interesting. The 2-cuts $\{v_0, v_1\}$ and $\{v_0, v_3\}$ are taken anyway, so if they are interesting, we took them in P_1. Otherwise, it does not matter: if there is an interesting 2-cut containing v_1 or v_3, it will be taken in another node. The case $k = 3$ is similar.

- If $k \leqslant 5$ but $G \neq C_k$ and there exists $i \in \{2, 3, \ldots, k-2\}$ such that the edges v_0v_1 and v_iv_{i+1} are virtual. Without loss of generality, we can consider that $i = 2$ and $k \in \{4, 5\}$. Consider the node connected to μ by the virtual edge v_0v_1. If it is not a P-node, name this node μ'_1. If it is a P-node, let μ'_1 be one of its neighbor different from μ (it exists as every P-node has degree at least 2 in T). Define similarly the node μ'_2 for the virtual edge v_iv_{i+1}. Let $v'_i \in V(\mu'_i) \setminus V(\mu)$. If $k = 5$, for similar reasons as in the last case, because of the existence of v'_1 and v'_2, the cuts $\{v_0, v_2\}$, $\{v_1, v_3\}$ and $\{v_1, v_4\}$ are interesting. Therefore, if $k = 5$, all vertices are taken in interesting 2-cuts, and we are done. If $k = 4$, the 2-cut $\{v_0, v_2\}$ (resp. $\{v_1, v_3\}$) is interesting if and only if $N[v_0] \not\subseteq N[v_2]$ or $N[v_2] \not\subseteq N[v_0]$ (resp. $N[v_1] \not\subseteq N[v_3]$ or $N[v_3] \not\subseteq N[v_1]$). If they are interesting, we took them, otherwise it does not matter: we have taken all possibly interesting 2-cuts. We are done with this case.

Therefore, the first property is proved. □

We say that T_i *displays* the vertices u and v through the node μ. A vertex that is part of an interesting minimal 2-cut but that is not displayed by T is called *hidden*. F *displays* u if at least one of the T_i's displays u.

Corollary IV.4.8. *Let G be a 2-connected graph, $S \subseteq V(G)$ and k be a constant depending on the graph. Suppose that for any interesting 2-cut tree, the number of vertices $u \in S$ that appear with some friend v such that $\{u, v\}$ is an interesting cut displayed by T is bounded by k. Then if C is the set of interesting vertices in 2-cuts, $|C \cap S| \leqslant 3k$.*

Bounding the Number of Interesting Vertices. We now can prove Lemma IV.2.2. We did not try to optimize the constants $c_{IV.2.2}(d)$ and $m_{IV.2.2}(\mathscr{C})$. Let f be the d-dimensional control function of the graph class. We prove the lemma for $c_{IV.2.2}(d) = 22 \cdot (d+1)$ and $m_{IV.2.2}(d) = f(11) + 4$.

Without loss of generality, one can assume that the graph is 2-connected. Indeed, if it is not, one can split G into 2-connected component and do the analysis on those components. Let T be a 2-cut tree of G, rooted at an arbitrary C node, D be a dominating set of $N^4[S]$ in G using vertices in $N^5[S]$. Let I be the set of interesting vertices displayed in T and $I' = I \cap S$. We prove this claim first.

Claim IV.4.9. $|I'| \leqslant 6 \cdot \mathrm{MDS}(G, N^4[S])$.

Let $u \notin D$ be an interesting vertex displayed in T and v be a friend of u, i.e. a vertex such that $c = \{u, v\}$ is a 2-cut with two components of $G - c$ not dominated entirely by v, and with $N[u] \not\subseteq N[v]$. We first prove the following claim:

Claim IV.4.10. *There exists $d \in D$ such that $d_G(u, d) \leqslant 5$ and d is lower in T than u. Moreover, the interesting-ness of a vertex is certified by vertices at distance at most 4.*

Proof of claim. Let μ a node of T below c that is contained in a component C of $G - c$ not fully dominated by v. Let $w \in C$ a vertex that is not dominated by v and that minimizes $d_G(u, w)$. w will be our witness of interesting-ness of one of the components. If one wishes to get the witness of another component C', one can apply the same technique on $w' \in C$, a vertex not dominated by v that minimizes $d_G(u, w')$. Note that w is well-defined by definition of C. Let us first prove that $d_G(u, w) \leqslant 4$. Note that $w \in C$ and therefore w is lower in T than u. If there exists some $x \in (N(u) \setminus N(v)) \cap C$, one can take $w = x$ and then $d_G(u, w) = 1$. Otherwise, take $y \in N(w)$ such that $d_G(u, y) < d_G(u, w)$. By minimality of $d(u, w)$, $y \in N(v)$. Take $x \in (N(u) \cap N(v)) \cap C$. Such an x always exists, because c is a minimal 2-cut (i.e. $N(u) \cap C \neq \varnothing$). The path $wyvxu$ exists, therefore $d(u, w) \leqslant 4$. One can choose a dominating vertex of D adjacent to w if $w \notin D$, or $d = w$ if $w \in D$. By the triangle inequality, $d(u, v) \leqslant 5$, and Claim IV.4.10 is proven. ◇

Let $q : I' \setminus D \to D$ a function that we define later as our charging function. Let $d \in D$ be a vertex of a node lower than u in T and such that $d(u, d) \leqslant 5$, chosen to be one of the highest in T among all possible candidates. Note that d is well-defined by Claim IV.4.10. We set $q(u) := d$ and say that u *charges* d. Now, we bound the size of the preimages of q.

For a fixed $d \in q(I' \cap S)$, choose $u \in q^{-1}(\{d\})$ highest in T. Again, by Proposition IV.4.7 there exists v a friend of u displayed in the same node as u. v can be chosen highest-in-T among all the possible candidates. Let $\mu_u := \{u, v\}$ and μ_d be the highest-in-T R-node that contains d. Notice that even though d may appear lower in T than μ_d, there cannot be any interesting vertex charging d lower than μ_d, as this would mean some interesting vertex charges a vertex higher than itself, which is not possible. Let μ' the lowest-in-T C-node that is higher than μ_d. Let F be the set of C-nodes of T that are between μ_u and μ' (both non-included), that form interesting 2-cuts, and such that one of the interesting vertices in the cut is in $S \setminus \{u\}$. There are two possible cases.

- Either there exists some $c \in F$ such that $c \cap N[v] = c \cap \mu' = \varnothing$. In this case, let w be the interesting vertex of c. We get a contradiction because w cannot be dominated by v nor a vertex below it, as d is lower than μ'.

- Therefore, all $c \in F$ either contain some $w \in \mu'$ or contain a vertex dominated by v. Notice that if w exists, it is unique for all $c \in F$.

Therefore, F is partitioned in two sets: F_2, the set of $c \in F$ that contain w, and F_1, the set of $c \in F \setminus F_2$ that contain some vertex dominated by v. Furthermore, all the $c \in F_1$ appear higher in the tree than the $c \in F_2$. We claim that $q^{-1}(\{d\}) \subseteq \mu_u \cup \mu' \cup \mu_c$ for some $\mu_c \in F \cup \{\varnothing\}$. Let us first prove that then all $c \in F_1$ but one 2-cut contain v. Let μ_c be the highest C-node of F_1 that does not contain v, if it exists. If it does not exist, set $\mu_c = \varnothing$. Suppose there exists a $c' \in F_1$ strictly below μ_c. All vertices of c' cannot be dominated by v. We are not in the first case, therefore c' must contain some vertex of μ', i.e. $c \in F_2$. Let $c \in F_1 \setminus \{\mu_c\}$ if μ_c exists, else let $c \in F_1$. Let $x \in q^{-1}(\{d\})$ be an interesting vertex in c different from u. x and its neighbors can only be dominated by v, therefore $N[x] \subseteq N[v]$ and x cannot be interesting, i.e. x does not exist. Similarly, let $c \in F_2$ and let $x \in q^{-1}(\{d\})$ be an interesting vertex in c that is not in $\mu_c \cup \mu'$, if it exists. x and its neighbors can only be dominated by w, therefore $N[x] \subseteq N[w]$ and x cannot be interesting, i.e. x does not exist.

We therefore get that $|q^{-1}(\{d\})| \leqslant 6 \cdot \mathrm{MDS}(G, N^4[S])$. This proves Claim IV.4.9. Moreover, by Corollary IV.4.8, we get the following claim.

Claim IV.4.11. *The number of interesting vertices of G in S is at most* $19 \cdot \mathrm{MDS}(G, N^4[S])$.

We can now get a bound on the number of interesting vertices in local 2-cuts.

Claim IV.4.12. *Let G be a graph of asymptotic dimension d with control function f. Then, the number of $(f(11)+5)$-local interesting vertices is bounded by* $22(d+1) \cdot \mathrm{MDS}(G)$.

This will directly imply the statement of the lemma. Fix $B \subseteq V(G)$ such that $G[B]$ has diameter $f(11)$. We claim the following:

Claim IV.4.13. *Every vertex in a $(f(11)+5)$-local 2-cut of G that is in B is also in a 1-cut or a 2-cut of $G[N^5[B]]$.*

Proof of claim. Indeed, let $v \in B$ such that $c = \{u, v\}$ is a $(f(11)+5)$-local 2-cut of G. Let $a, b \in N(v) \setminus c$ two distinct vertices separated by v, i.e. there are not two internally-disjoint ab-paths in $G[N^{f(11)+5}[c]] - c$. Notice that $N[B] \subseteq N^{f(11)+5}[c]$ because B has weak diameter at most $f(11)$. ab are separated by c in $G[N[B]]$, because $a, b \in N[B]$ and because no two disjoint ab-paths in $N[B] \subseteq c$ exist. Therefore, Claim IV.4.13 is proven. ◇

Every interesting vertex of B is either a 1-cut of $G[N^5[B]]$ or in a 2-cut of $G[N^5[B]]$ and interesting in $G[N^5[B]]$. Indeed, by Claim IV.4.10, the interesting-ness of a vertex is certified by a vertex at distance at most 4.

By Claim IV.4.2, we can bound the number of vertices in 1-cuts of $G[N^5[B]]$ in B by $3\,\mathrm{MVC}(G[N^5[B]], N[B]) \leqslant 3\,\mathrm{MVC}(G, N[B])$, and by

Claim IV.4.11, we can bound the number of interesting vertices of 2-cuts of $G[N^5[B]]$ in B by $19\,\mathrm{MVC}(G[N^5[B]], N^4[B]) \leqslant 19\,\mathrm{MVC}(G, N^4[B])$.

By the definition of the asymptotic dimension, there exists a cover $B_0, B_1, \ldots, B_d$ of G where the 11-components (connected components of G^11) of B_i have weak diameter at most $f(11)$. Note that each B_i contains distinct 11-components which are of distance at least 12 from each other. With an abuse of notation, when we write $B \in B_i$ we mean that B is a 11-component of B_i. That is, we treat B_i as the set of its 11-components. As $B_0, B_1, \ldots, B_d$ is a cover of $V(G)$ by subsets of diameter at most $f(11)$, we can bound the number of $(f(11)+5)$-local 2-cuts of G by summing the number of 2-cuts of all the $G[N^4[B_i]]$'s. Let I be the set of interesting vertices in $(f(11)+5)$-local 2-cuts of G. We get

$$|I| \leqslant \sum_{i=0}^{d} \sum_{B \in B_i} \left(3 \cdot \mathrm{MDS}(G, N[B]) + 19 \cdot \mathrm{MDS}(G, N^4[B])\right) .$$

Notice that because the B_i's are partitioned into their 11-components, all elements of $\{N^5[B] \mid B \text{ is a 11-component of } B_i\}$ are pairwise disjoint. Therefore, by Lemma IV.4.1, we get

$$|I| \leqslant \sum_{i=0}^{d} 22 \cdot \mathrm{MDS}(G) = 22(d+1) \cdot \mathrm{MDS}(G) .$$

This finishes the proof of Lemma IV.2.2.

IV.4.3 *Proof of Lemma IV.3.2: Brute-forcing is fast enough*

To prove this, we need a few results from [Din17].

The structure of 3-connected $K_{2,t}$-minor-free graphs. Let us give a quick overview of their result. The author defines two types of graphs. The first type is a generalization of outerplanar graphs. Let G be a graph with a specified Hamiltonian cycle C, which is called the reference cycle. Edges of G that are not part of C are called chords. Two non-incident chords ab and cd are said to cross if the vertices a, c, b, d appear in that order around C. G is said to be of type-I if each chord crosses at most one other chord. Additionally, if two chords ab and cd do cross, then either both ac and bd are edges of C, or both ad and bc are edges of C. The class of all type-I graphs is denoted by $\mathcal{P}$.

Let H be a type-I graph with reference cycle C, and let ab and cd be two distinct edges of C. Assume all chords of C lie between the two paths of $C \setminus \{ab, cd\}$. If ab and cd share an endpoint, say $a = d$, then H is called a fan with corners a, b, c. The vertex a is called the center of the fan, and the number of chords is called the fan's length. If ab and cd have no shared endpoints, then for any subset $F \subseteq \{ab, cd\}$, $H \setminus F$ is called a strip with corners a, b, c, d, provided the minimum degree of $H \setminus F$ is at least two. The radius of a strip H as $\max\{d_H(h, y) \mid h \in H, x \in \{a, b, c, d\}\}$.

Strips are proven to be $K_{2,5}$-minor-free in [Din17], and it is not hard to see that if the radius of a strip is large, then its corners form local cuts.

Let G be a graph. Adding a fan or a strip to G means identifying the corners of a fan or a strip (which is disjoint from G) with distinct vertices of G. An augmentation of G is obtained by adding disjoint fans and strips to G, with the condition that if two corners are identified with the same vertex of G, then one of them is the center of a fan, and the other is either a center of a fan or a corner of a strip. For any integer m, $\mathscr{B}_m$ is defined as the class of graphs on at most m vertices, and A_m is the class of all augmentations of graphs in $\mathscr{B}_m$.

> **Proposition IV.4.14** (Corollary 1.6 of [Din17]). *Let t be an integer and $\mathscr{C}_t$ be the class of $K_{2,t}$-minor-free graphs. Then there exists some $m_{IV.4.14}(t)$ such that $\mathscr{C}_t \subseteq \mathsf{A}_{m(t)}$.*

Embedding components into a 3-connected graph. In the following, we prove Lemma IV.3.2.

Already, note that our algorithm takes all 1-cuts. We can therefore assume that the rest of the graph that remains to be solved is 2-connected: if C_1 is the set of all 1-cuts of G, let $V_1, V_2, \ldots, V_k$ be the different connected components of $G - C_1$. Consider for every $i \in \{1, 2, \ldots, k\}$, $G_i := G[V_i \cup N[V_i] \cap C_1]$ and do the analysis on this graph. If there exists a constant approximation algorithm for 2-connected graphs, then there exists a constant approximation algorithm for 1-connected graphs. We want to do a similar reasoning for 2-cuts. However, our algorithm does not take all 2-cuts, but only interesting vertices of these 2-cuts.

In G, let X be the set of $m_{IV.3.2}(t)$-local 1-cuts, X_2 be the set of vertices inside an $m_{IV.2.2}(\mathscr{C}_t)$-local 2-cuts, I the set of $m_{IV.2.2}(\mathscr{C}_t)$-important C_1 the set of 1-cuts, and C_2 the set of 2-cuts. Finally, let $Y = X \cup I$ and U be the set of vertices in $\{u \in N[Y] \mid N[u] \subseteq N[Y]\}$ the set of vertices dominated by Y that do not have neighbors not dominated by Y.

Let C be a connected component of $G - (Y \cup U)$. We want to embed $G[C]$ into a 3-connected graph G' without creating any $K_{2,t}$ minor in G'. Let $G_1 = G[C \cup (N[C] \cap Y)]$. We add the edges $\{u, v\}$ that are formed by contracting the connected components of $G - G_1$. We claim that G_1 is 3-connected. Let c be a 1-cut or 2-cut of G_1, separating it into at least two connected components A and B. It cannot be that $c \subseteq Y \cup U$, as C is connected and would not intersect c. Therefore, c is not a 1-cut of G, nor an interesting cut. If it is not a 2-cut, there exists an AB-path in G that does not intersect c, and this path would still exist after the contraction of the definition of G_1, contradiction. If c is a 2-cut of G, one of the vertices v of $c = \{u, v\}$ is not interesting. There are two possibilities: either $N[v] \subseteq N[u]$, or all connected components of $G - c$ but one (assuming G is connected) are dominated entirely by u. In the first scenario, there are two subcases. If $u \in Y$, then $v \in U$ and $c \subseteq Y \cup U$, contradiction. Otherwise, $u \notin Y$. Note that $N[u] \not\subseteq N[v]$ as u and v would be true twins. v has to dominate all but one component of $G - c$. Then this also applies for u, and we enter the second scenario. In the second scenario, delete all vertices that are in connected components of $G - c$ that are dominated entirely by u, and add the edge uv if it does not exist. Again, this creates a graph with diameter at least one less, and this does not create any new 2-cuts. The resulting graph G' is 3-connected, as all 2-cuts from G' have been handled.

Bounding the diameter of components. By Proposition IV.4.14, there exists some $m_{IV.4.14}(t)$ such that G' is an augmentation of some graph $\overline{G'}$ of bounded size, to which strips and fans are added. We would like to say that G' has bounded radius. $\overline{G'}$ and the attached fans have bounded radius, and we only have to bound the size of the strips of G'. Note that if some strip has radius at least $3m_{IV.2.2}(\mathscr{C}_t)$, its corners in $\overline{G'}$ form two $m_{IV.2.2}(\mathscr{C}_t)$-local 2-cuts c_1 and c_2 in G'. Those cuts are also $m_{IV.2.2}(\mathscr{C}_t)$-local 2-cuts in G, because G' is an induced minor of G: suppose that c is a local cut in G' but not in G. There would be a short path in G between two different connected components of $G' - c$. However, as G' is an induced minor of G, the path still exists and can only be shorter. Furthermore, c_1 and c_2 are both in $Y \cup U$. Suppose they are not. Let $c_1 = \{u, v\}$. As the strip has a long radius, it cannot be that $N[u] \subseteq N[v]$ or that $N[v] \subseteq N[u]$. Without loss of generality, u dominates all but one connected component of $G - c$. This means that u is connected to some vertex on the other end of its own strip, which is impossible. Finally, G' cannot include strips of radius at least $3m_{IV.2.2}(\mathscr{C}_t)$. Therefore, G' has radius bounded by $3m_{IV.2.2}(\mathscr{C}_t) + m_{IV.4.14}(t)$, C also has radius bounded by $m_{IV.3.2}(t) = 3m_{IV.2.2}(\mathscr{C}_t) + m_{IV.4.14}(t) + 3$.

IV.4.4 *Proof of Theorem IV.3.4: A linear approximation in constant rounds*

The goal of this subsection is to build a simple $(2t-1)$-approximation algorithm that has constant round complexity (that does not depend on t) in $K_{2,t}$-minor-free graphs.

We need the following result from Ore.

Lemma IV.4.15 ([Ore62])**.** *Every n-vertex graph without isolated vertices (a vertex with no adjacent edges) has a dominating set of size at most $\frac{n}{2}$.*

When some graph G is considered, $\gamma(v)$ will denote the minimum number of vertices different from v needed to dominate $N[v]$. We will also denote the set of vertices whose neighborhood cannot be dominated by less than two vertices (other than the original vertex) by $D_2(G) = \{v \in V(G) \mid \gamma(v) \geqslant 2\}$.

Lemma IV.4.16. *Let G a graph, $S \subseteq V$ be a set of vertices, and $D \subseteq N^2[S]$ a minimum dominating set of $N[S]$ in G of size k. Then there exists some minor H of $G[N^2[S]]$ such the following hold:*

1. *$V(H) = A \sqcup B$ and $|B| = k$,*
2. *$H[A]$ is edgeless and $d_H(a) \geqslant 2$ for all $a \in A$ ($d_H(v)$ is the degree of v in H),*
3. *and $|A| \geqslant \frac{1}{2}|(D_2 \cap S) \setminus D|$.*

Proof. Let $G' = G[N^2[S]]$. Let us start by constructing a minor as in Lemma 1 of [KSV21]. Take D a dominating set of size k, and write $D = \{d_1, d_2, \dots, d_k\}$. Define H as the minor of G' by contracting the branch sets b_i defined as follows.

$$b_i = N_{G'}[d_i] \setminus (D_2 \setminus D \cup \bigcup_{j<i} N[d_i] \cup \{d_{i+1}, \dots, d_k\})$$

Let $B = \{b_1, b_2, \dots, b_k\}$ and $A = (D_2 \cap S) \setminus D$. For any $v \in (D_2 \cap S) \setminus D$, we have $d_H(v) \geqslant 2$, as D is a dominating set of $N[S]$. Here we need an additional trick: we remove edges uv in triangles of the form u, v, d where $u, v \in (D_2 \cap S) \setminus D$ and $d \in D$. This is always possible while keeping $d_H(v) \geqslant 2$ for any $v \in (D_2 \cap S) \setminus D$. Indeed, D is a dominating set of G and $\gamma(v) \geqslant 2$. Since $\gamma(v) \geqslant 2$, there exist two paths in G of length at most 2 from v to distinct vertices of D. Moreover, these two paths only intersect at v. One can check that these paths also exist in H. We now want to contract some edges so that every non-contracted vertex left in $(D_2 \cap S) \setminus D$ is adjacent to two vertices of B in H. Consider the induced subgraph $H[(D_2 \cap S) \setminus D]$. Let I be its set of isolated vertices and $J = (D_2 \cap S) \setminus (D \cup I)$ the rest of the vertices. Every vertex of I is adjacent to two vertices of B in H. However, it is not the case for vertices in J. By definition, $H[J]$ is a graph with no isolated vertices. Let D' be a minimum dominating set of $H[J]$, which is of size at most $\frac{1}{2} \cdot |J|$ by Lemma IV.4.15. Every vertex $j \in J$ is adjacent to some b_{k_j} (because D is a dominating set). Contract the edges jb_{k_j} for $j \in D'$, i.e. for every $j \in D'$, add the vertex j to the branch set b_{k_j} and remove j from A. Because D' is a dominating set of J, every vertex in $J \setminus D'$ is now adjacent to two vertices of B in H. Moreover, because $|D'| \leqslant \frac{1}{2} \cdot |J| \leqslant \frac{1}{2} \cdot |(D_2 \cap S) \setminus D|$, we have $|A| \geqslant \frac{1}{2} \cdot |(D_2 \cap S) \setminus D|$. Delete all the edges from $H[A]$. This ends the proof. □

Using the same notation as in the above lemma, the next lemma bounds the size of A with respect to $|B|$ on $K_{2,t}$-minor-free graphs.

Lemma IV.4.17. *Let $G = (A \sqcup B, E)$ a $K_{2,t}$-minor-free graph such that $G[A]$ is edgeless and every vertex $v \in A$ has degree at least 2. Then $|A| \leqslant (t-1)|B|$.*

Proof. First, apply some preprocessing to the graph: if there exists some $a \in A$ and $b, b' \in N(a)$ such that b and b' are not connected in $G[B]$, then we can contract the edge ab, and from now on consider the newly contracted graph instead of G. This creates a new "red" edge bb'. Put it in R, the set of red edges. Thinking about red edges as paths of length 2 allows us to do a fine-grain analysis of the size of $|A|$. Keep repeating this step whenever some such a exists, and let $G' = (A' \sqcup B, E')$ be the final graph obtained after this procedure.

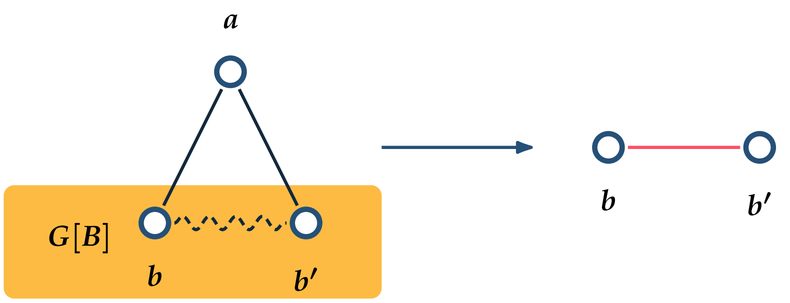


Figure IV.1: An explanation of the preprocessing procedure. Here the dashed squiggly edge represents the lack of path in $G[B]$.

We prove by induction on $|B|$ that $|A'| + |R| \leqslant (t-1)|B|$.

The base case with $B = \varnothing$ is trivial. If B is not empty, let C be a connected component of $G'[B]$. We can find some $c \in C$ such that $G'[C \setminus \{c\}]$ is connected (for instance, one can take a leaf in a BFS tree of $G'[C \setminus \{c\}]$). Let $R(c) := \{cb \in E' \mid b \in V(B)\} \cap R$ denote the set of red edges that touch c, i.e. the red edges that are not in $G'[B \setminus \{c\}]$. Let $a \in N(c) \cap A'$ and $b \in N(a) \setminus \{c\}$. Then $b \in C$, i.e. b is in the same connected component of $G'[B]$ as c.

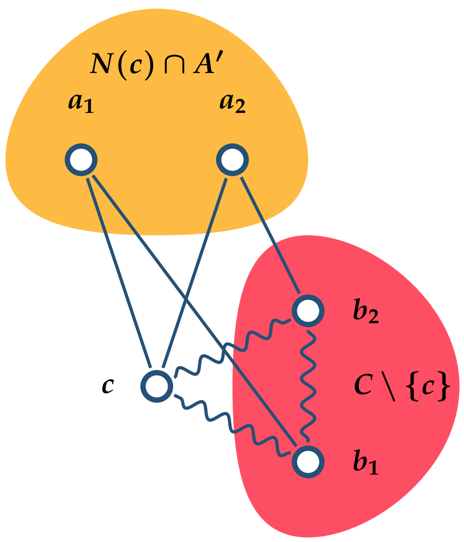


Figure IV.2: The vertices at the top are in A' and the vertices on bottom are in B. In the yellow region, the neighbors a_i of c in A', which are all connected to a vertex $b_i \in C \setminus \{c\}$ in the red region. c and the b_i are all in the same connected component of $G[B]$.

Thus, we have the bound $|N(c) \cap A'| + |R(c)| \leqslant t-1$, as otherwise we would obtain a $K_{2,t}$ minor by contracting the vertex set $C \setminus \{c\} \neq \varnothing$. Remove the vertex c from B and the vertices $N(c) \cap A'$ from A', to maintain the condition on the vertex degrees of A'. Notice that for each $a \in A' \setminus N(c)$, if $b, b' \in N(a)$ then b and b' are still connected in $G'[B]$. Indeed, if they were connected before but not anymore, we would have $b, b' \in C$, but as $C \setminus \{c\}$ is connected, we would get a contradiction. This new smaller graph contains all the red edges except the ones in $R(c)$. We have removed at most $t-1$ vertices from A' or edges from R, so by applying induction hypothesis we have that $|A'| + |R| \leqslant (t-1)|B|$. Finally, $|A| = |A'| + |R| \leqslant (t-1)|B|$. $\square$

Moreover, we prove that in a graph with no true twins, D_2 is a dominating set.

Lemma IV.4.18. *Let G be a graph with no true twins and let $v \in V(G)$ such that $\gamma(v) = 1$. Then there exists some $u \in V(G)$ such that $\gamma(u) \geqslant 2$ and $N[v] \subseteq N[u]$.*

Proof. Let $v \in V(G)$ such that $\gamma(v) = 1$. Therefore, there exists some u such that $N[v] \subsetneq N[u]$ because G contains no true twins. Take u such that $N[u]$ is maximal. Then $\gamma(u) \geqslant 2$. Indeed, if we had $\gamma(u) = 1$, there would be some other vertex u' such that $N[u] \subsetneq N[u']$, because G does not contain true twins. Contradiction. □

Using the preceding lemmas, we get the following.

Corollary IV.4.19. *Let G be a graph, and $S \subseteq V(G)$ a subset of vertices such that $G[N^2[S]]$ is $K_{2,t}$-minor free. Then $|D_2(G) \cap S| \leqslant (2t-1)\,\mathrm{MDS}(N[S])$.*

Proof. Let H be the minor defined in Lemma IV.4.16. Apply Lemma IV.4.17 on H. We get that $|D_2(G) \setminus D| \leqslant 2|A| \leqslant 2(t-1)\,\mathrm{MDS}(G)$, so $|D_2(G)| \leqslant \mathrm{MDS}(G) + |D_2(G) \setminus D| \leqslant (2t-1)\,\mathrm{MDS}(G)$. □

We can now prove that the following is a $(2t-1)$-approximation algorithm in $K_{2,t}$-minor-free graphs.

1. Make the graph without true twins. u and v are true twins in G if $N[u] = N[v]$. The true-twin-less graph associated to G is the largest subgraph of G with no true twins.

2. Return the set $D_2 = \{v \in V(G) \mid \nexists u \in V(G-v), N[v] \subseteq N[u]\}$. It is the set of all vertices whose neighborhood cannot be dominated by a single vertex other than itself.

The round complexity of this algorithm is constant, and does not depend on t. Moreover, from Corollary IV.4.19 and the fact that the true-twin-less graph associated to G has the same domination number as G, we get that this is indeed a $(2t-1)$-approximation algorithm. This proves Theorem IV.3.4.

References

[ASS19] Saeed Akhoondian Amiri, Stefan Schmid, and Sebastian Siebertz. “Distributed Dominating Set Approximations beyond Planar Graphs”. In: *ACM Transactions on Algorithms* 15.3 (2019), Article No. 39, pp. 1–18. DOI: 10.1145/3093239.

[BBE+24] Marthe Bonamy et al. "Asymptotic Dimension of Minor-Closed Families and Assouad–Nagata Dimension of Surfaces". In: *Journal of the European Mathematical Society* 26.10 (2024), pp. 3739–3791. DOI: 10.4171/JEMS/1341.

[BCGW21] Marthe Bonamy et al. "A Tight Local Algorithm for the Minimum Dominating Set Problem in Outerplanar Graphs". In: *35th International Symposium on Distributed Computing* (*DISC*). Ed. by Seth Gilbert. Vol. 209. LIPIcs. Paris, France, Sept. 2021, 13:1–13:18. DOI: 10.4230/LIPIcs.DISC.2021.13.

[BGPW25a] Marthe Bonamy et al. "Distributed Approximation Algorithms for Minimum Dominating Set in Locally Nice Graphs". In: *arXiv preprint arXiv:2507.04960* (2025).

[BGPW25b] Marthe Bonamy et al. "Local Constant Approximation for Dominating Set on Graphs Excluding Large Minors". In: *44th Annual ACM Symposium on Principles of Distributed Computing* (*PODC*). Huatulco, Mexico: ACM Press, June 2025. DOI: 10.1145/3732772.3733531.

[CHW18] Andrzej Czygrinow, Michał Hańćkowiak, and Wojciech Wawrzyniak. "Distributed Approximation Algorithms for the Minimum Dominating Set in K_h-Minor-Free Graphs". In: *29th International Symposium on Algorithms and Computation* (*ISAAC*). Vol. 123. LIPIcs. Jiaoxi, Yilan, Taiwan, Dec. 2018, 22:1–22:12. DOI: 10.4230/LIPIcs.ISAAC.2018.22.

[Din17] Guoli Ding. *Graphs without large $K_{2,n}$-minors*. 2017. DOI: 10.48550/ARXIV.1702.01355. URL: https://arxiv.org/abs/1702.01355.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[HKO+25] Ozan Heydt et al. "Distributed domination on sparse graph classes". In: *European Journal of Combinatorics* 123 (Jan. 2025), p. 103773. DOI: 10.1016/j.ejc.2023.103773.

[HLS14] Miikka Hilke, Christoph Lenzen, and Jukka Suomela. "Brief Announcement: local approximability of minimum dominating set on planar graphs". In: *33rd Annual ACM Symposium on Principles of Distributed Computing* (*PODC*). Paris, France: ACM Press, July 2014, pp. 344–346. DOI: 10.1145/2611462.2611504.

[KMW16] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. "Local Computation: Lower and Upper Bounds". In: *Journal of the ACM* 63.2 (Mar. 2016), Article No. 17, pp. 1–44. DOI: 10.1145/2742012.

[Kos84] Alexandr V. Kostochka. "A lower bound for the Hadwiger number of graphs by the average degree". In: *Combinatorica* 4.4 (Dec. 1984), pp. 307–316. DOI: 10.1007/BF02579141.

[KP10] Alexandr V. Kostochka and Noah Prince. "Dense graphs have $K_{3,t}$ minors". In: *Discrete Mathematics* 310.20 (Oct. 2010), pp. 2637–2654. DOI: 10.1016/j.disc.2010.03.026.

[KSV21] Simeon Kublenz, Sebastian Siebertz, and Alexandre Vigny. "Constant Round Distributed Domination on Graph Classes with Bounded Expansion". In: *28th International Colloquium on Structural Information & Communication Complexity (SIROCCO)*. Ed. by Tomasz Jurdziński and Stefan Schmid. Vol. 12810. Lecture Notes in Computer Science. Wrocław, Poland: Springer, Cham, June 2021, pp. 334–351. DOI: 10.1007/978-3-030-79527-6_19.

[KYK80] Tohru Kikuno, Noriyoshi Yoshida, and Yoshiaki Kakuda. "The NP-Completeness of the Dominating Set Problem in Cubic Planar Graphs". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E63-E.6 (1980), pp. 443–444.

[LPW13] Christoph Lenzen, Yvonne-Anne Pignolet, and Roger Wattenhofer. "Distributed minimum dominating set approximations in restricted families of graphs". In: *Distributed Computing* 26.2 (Mar. 2013), pp. 119–137. DOI: 10.1007/s00446-013-0186-z.

[LW24] Christoph Lenzen and Sophie Wenning. *Tight Bounds for Constant-Round Domination on Graphs of High Girth and Low Expansion*. Tech. rep. 2408.12998v1 [cs.DC]. arXiv, Aug. 2024. DOI: 2408.12998v1.

[Ore62] Øystein Ore. *Theory of Graphs*. Amer. Math. Soc., 1962, pp. 206–212.

[RW16] Bruce A. Reed and David R. Wood. "Forcing a sparse minor". In: *Combinatorics, Probability and Computing* 25.2 (Mar. 2016), pp. 300–322. DOI: 10.1017/S0963548315000073.

[Sol19] Ryan Solava. "On the fine structure of graphs avoiding certain complete bipartite minors". PhD thesis. Vanderbilt University, 2019.

[Suo13] Jukka Suomela. "Survey of Local Algorithms". In: *ACM Computing Surveys* 45.2 (Feb. 2013), Article No. 24, pp. 1–40. DOI: 10.1145/2431211.2431223.

[Tho01] Andrew Thomason. "The Extremal Function for Complete Minors". In: *Journal of Combinatorial Theory, Series B* 81.2 (Mar. 2001), pp. 318–338. DOI: 10.1006/jctb.2000.2013.

[Waw14] Wojciech Wawrzyniak. "A strengthened analysis of a local algorithm for the minimum dominating set problem in planar graphs". In: *Information Processing Letters* 114.3 (Mar. 2014), pp. 94–98. DOI: 10.1016/j.ipl.2013.11.008.

Chapter V
Lower bound and a conjecture

Abstract

We conjecture that by allowing the number of rounds to be an arbitrary function of H, we can obtain an approximation ratio within constant factor of the *pathwidth* of H. We provide a matching lower bound that guarantees that it would be optimal if true.

Acknowledgements

This section is part of a work with Marthe Bonamy, Cyril Gavoille, Tim Planken, and Alexandra Wesolek, adapted to the thesis. The work of this chapter is not yet published. The continuation of this work can be found in Chapter VIII.

Contents

V.1 Introduction

In Chapter IV, we proved the following theorem.

Theorem IV.3.1. *For every integer $t \geqslant 2$, Algorithm 5 is an $O_t(1)$-round 50-approximate deterministic* LOCAL *algorithm for* Minimum Dominating Set *on $K_{2,t}$-minor-free graphs.*

One obvious question that comes to mind is: if we replace $K_{2,t}$ in the statement of Theorem IV.3.1 by $K_{s,t}$ (for some small $s \leqslant t$), what would be the right approximation factor with respect to s, if it is even bounded independently of t? We first prove a lower bound on the approximation factor for such an algorithm. More specifically, it has to be at least a linear function of s. Interestingly, the approximation factor lower-bound only depends on the *pathwidth* of the forbidden minor H, which measures how much H resembles a path. The *pathwidth* of H, denoted as $\mathrm{pw}(H)$, is the smallest integer p such that H is a subgraph of some interval graph with maximum clique size $p+1$. For a more formal definition and in-depth introduction to pathwidth, see Section VI.1.

Theorem V.1.1. *For every graph H and every $\varepsilon > 0$, there is no $O_H(1)$-round $(2 \cdot \mathrm{pw}(H) - 1 - \varepsilon)$-approximation deterministic* LOCAL *algorithm for* Minimum Dominating Set *on H-minor-free graphs.*

Note that all the algorithms are deterministic, since in general constant-round randomized LOCAL algorithms are not possible if high probability guarantee is required. Our lower bound applies also to deterministic LOCAL algorithms.

Noticing that $K_{s,t}$ has pathwidth s, Theorem V.1.1 shows that one cannot hope for an $o(s)$-approximation on $K_{s,t}$-minor-free graphs. Together with Theorem IV.3.1, the lower bound inspires the following conjecture, which would provide a deeper truth behind the behavior of MDS in H-minor-free graphs.

Conjecture V.1.2. *For every graph H, there is an $O(\mathrm{pw}(H))$-approximation $O_H(1)$-round* LOCAL *algorithm for* Minimum Dominating Set *on H-minor-free graphs.*

Whereas Conjecture V.1.2 claims that an $O(\mathrm{pw}(H))$-approximation exists for H-minor-free graphs, the best current upper bound on the approximation ratio is only $t^{O(pt\sqrt{\log p})}$, where $t = |V(H)|$ and $p = \mathrm{pw}(H)$. Conjecture V.1.2, if true, implies in particular that there is an $O(s)$-approximation $O_t(1)$-round LOCAL algorithm for $K_{s,t}$-minor-free graphs (for $s \leqslant t$). This corroborates with the results of Chapter III, where we exhibit a constant round and approximation for genus-g graphs (which are $K_{3,2g+3}$-minor-free) whose approximation ratio is independent of g Algorithm 2, and with the results of Chapter IV, as $K_{2,t}$ has pathwidth

2. We prove the case where $\mathrm{pw}(H) = 2$ in Chapter VIII by extending the techniques of Chapter IV. Unfortunately, there is no hope for an immediate generalization of this result to a constant factor approximation for all H.

Graph class	Approx. ratio upper	Approx. ratio lower	#rounds	References upper	References lower
trees, $p = 2$	3	$3 - \varepsilon$	2	Folklore [1]	[BCGW21]
outerplanar, $p = 2$	5	$5 - \varepsilon$	2	[BCGW21]	
planar, $p = 3$	$11 + \varepsilon$	$7 - \varepsilon$	$O_\varepsilon(1)$	[HKO+25]	[HLS14]

Table V.1: Constant-round approximation distributed algorithms for Minimum Dominating Set on some classes of graphs. Here p denotes the smallest pathwidth of a forbidden minor of the class. All the lower bounds hold for an arbitrary number of rounds, possibly depending on $|V(H)|$.

In fact, as suggested by Section V.1, we could hope for the following far bolder conjecture:

Conjecture V.1.3. *For every graph H, there is a $(2 \cdot \mathrm{pw}(H) + 1)$-approximation $O_H(1)$-round* LOCAL *algorithm for* Minimum Dominating Set *on H-minor-free graphs.*

Note that Conjecture V.1.3 would notably improve the state of the art for planar graphs[2], and that even base cases seem currently out of reach. For example, it would imply that for any graph H which is a collection of paths with an arbitrary number of leaves on all their vertices[3], we can design a 3-approximation algorithm in $O_H(1)$ rounds for MDS in H-minor-free graphs. This does not seem straightforward even for H being a star.

The suspicious reader could note that when a graph H has pathwidth at most 2, the graph H is necessarily planar. Therefore, H-minor-free graphs have bounded treewidth. One could be tempted to think the right generalization of Theorem IV.3.1 is not Conjecture V.1.2, but rather that graphs of bounded treewidth admit a constant approximation (not depending on the treewidth) as long as the number of rounds is allowed to be an arbitrary function of the treewidth. This is unfortunately completely hopeless, as the class of graphs forbidding a large balanced binary tree has bounded treewidth, yet one can observe that such a tree has arbitrarily large pathwidth, hence Theorem V.1.1 guarantees that the approximation ratio for such a class needs to depend on the treewidth.

[1] If there are at least three vertices, take all vertices with degree at least two, cf. [LPW13, BCGW21]. This requires two rounds from the model, because the vertices do not know their degree and need one round to count their neighbors (by counting the number of received messages).

[2] Note that planar graphs have no $K_{3,3}$-minor, and that $\mathrm{pw}(K_{3,3}) = 3$, suggesting a matching bound of 7 for the approximation ratio.

[3] A.k.a. caterpillar forest.

V.2 Lower Bound and Theorem V.1.1

Recall that the pathwidth of a graph H, denoted by $\mathrm{pw}(H)$, is the smallest p such that H is a subgraph of an interval graph whose maximum clique is $p + 1$. Obviously, $\mathrm{pw}(H) < |V(H)|$.

A rough upper bound of $t^{O(t^2\sqrt{\log t})}$ on the approximation ratio can be derived from [KSV21], using the fact that H-minor-free graphs are also K_{t+1}-minor-free graphs, $t = |V(H)|$.

A direct consequence of Theorem V.1.1 is that K_t-minor-free graphs or T_t-minor-free graphs, where T_t is a depth-t complete binary tree, have no $o(t)$-approximation $f(t)$-round LOCAL algorithm for MDS. This is because $\mathrm{pw}(K_t) = t-1$ and $\mathrm{pw}(T_t) = \lceil t/2 \rceil$ [Bod98, Th. 67]. Indeed, pathwidth is closed under taking minors, so graphs of pathwidth t are K_{t+2}-minor-free and T_{2t+1}-minor-free.

What makes lower bounds not so trivial in the LOCAL model is that the algorithm can benefit from the numerical values of the identifiers. Ideally, we would like to argue that this does not really help the algorithm. Unfortunately, this feature of the model helps in many ways, and the LOCAL version of the Cove-Vishkin's parallel algorithm gives a brilliant demonstration of this: it is possible to properly 3-color an n-vertex ring in only $O(\log^* n)$ rounds[4] [CV86]. This is tight from Linial's beautiful proof [Lin92]: $\frac{1}{2}(\log^* n - 1)$ rounds are needed to 3-color a ring. In fact, for infinitely many n, exactly $\frac{1}{2}\log^* n$ rounds are necessary and sufficient [RS15].

[4] Here $\log^*(n)$ is the minimum integer i such that $\log^{(i)} n \leqslant 1$, where $\log^{(i)} n$ is the i-th iterate base-2 logarithm of n.

As some previous lower bounds, our proof relies on Ramsey's Theorem, but perhaps in a more direct way. According to [CHW08], the elegant use of Ramsey's Theorem in LOCAL algorithms is due to [Pan99], in the context of Coloring. Using a strategy of repeated applications of Ramsey's Theorem [CHW08] proved a lower bound on the approximation ratio for Independent Set, and also for MDS using a reduction from Independent Set. [HLS14] use a similar strategy for the lower bound of $7-\varepsilon$ for approximating MDS in planar graphs.

The proof of Theorem V.1.1 is actually a direct consequence of the following Lemma V.2.1. For every integer $t \geqslant 1$, let $\mathscr{C}_t$ be the family of graphs composed of the t-th power of a path[5] in which extra vertices of degree one are attached to one extremity of the path. We assume as usually in the LOCAL model that for each graph $G \in \mathscr{C}_t$, the algorithm running on G uses identifiers from $\{1, \ldots, |V(G)|\}$ and can exploit the identifiers in arbitrary ways.

[5] More precisely, the i-th and j-th vertex of the path are connected if $|i-j| \leqslant t$.

Lemma V.2.1. *For every t and every function f, every $f(t)$-round deterministic LOCAL algorithm that approximates* Minimum Dominating Set *on $\mathscr{C}_t$ has approximation ratio at least $2t+1-\varepsilon$, for every $\varepsilon > 0$.*

We observe that all graphs of $\mathscr{C}_1$ are trees and all graphs of $\mathscr{C}_2$ are outerplanar graphs. So Lemma V.2.1 gives an alternative proof for the $3-\varepsilon$ and $5-\varepsilon$ known lower bounds for trees and outerplanar graphs. It is also easy to check that $\mathscr{C}_t$ is K_{t+2}-minor-free and $K_{t,t+1}$-minor-free, $t > 1$. The proof of Lemma V.2.1 is available in the Section V.3.

*Proof of **Theorem V.1.1**.* It is not difficult to see that all the graphs of $\mathscr{C}_t$ have pathwidth at most t. In particular, each graph of $\mathscr{C}_t$ excludes any graph of pathwidth $t+1$ as a minor. In other words, for each graph H of pathwidth at least two, $\mathscr{C}_{\mathrm{pw}(H)-1} \subseteq H$-minor-free. Therefore, one can apply Lemma V.2.1 with $t = \mathrm{pw}(H)-1$ to complete the proof of Theorem V.1.1. □

V.3 Proof of Lemma V.2.1: Lower bound for minor-free graphs

We prove a lower bound for constant round LOCAL algorithms for MDS for the class $\mathscr{C}_t$ which are t-th powers of paths with additional leaves. Usually lower bounds have been given by a repetitive use of Ramsey's theorem; see [HLS14]. This proof technique gives a lower bound on $\mathscr{C}_t$. However, we can use the extra leaves to simplify the proof to one single use of Ramsey's theorem.

Lemma V.2.1. *For every t and every function f, every $f(t)$-round deterministic LOCAL algorithm that approximates* Minimum Dominating Set *on $\mathscr{C}_t$ has approximation ratio at least $2t+1-\varepsilon$, for every $\varepsilon > 0$.*

Proof. A p-set is simply a set with p elements. We are interested in coloring p-sets with k colors. The well-known Ramsey Theorem states that, for every $k, p, q \in \mathbb{N}$ where $1 \leqslant k \leqslant p \leqslant q$, there is a smallest integer n, denoted by $R_k(p,q)$, such that, for every k-coloring of all p-sets of $\{1,\ldots,n\}$, there exists a q-set of $\{1,\ldots,n\}$ for which all its p-sets have the same color. It is well-known, from [ES35, Eq. (3), p. 466], that $R_2(p,q) \leqslant \binom{p+q-2}{p-1}$. See [GNNW24] for recent progress.

Let $r = f(t)$, consider any r-round deterministic LOCAL algorithm A for Minimum Dominating Set in $\mathscr{C}_t$, and let α be its approximation ratio. We will concentrate our attention to the graphs $P \in \mathscr{C}_t$ whose vertex-id's are in increasing order w.r.t. the underlying path of P (before taking the t-th power).

Consider such a graph $P \in \mathscr{C}_t$ with $N = |V(P)|$ and whose underlying path has $n \leqslant N$ vertices. We assume that N and n are suitable and large enough integers, namely $N \geqslant R_2(2rt+1, n-rt)$, $n \geqslant 2r(t+1)+1$, and $2t+1$ divides $n+t$. In fact, in order to prove that $\alpha \geqslant 2t+1-\varepsilon$, we will need that $n = \Omega(rt^2/\varepsilon)$. Let $x_1 < x_2 < \cdots < x_n$ be the IDs of its path vertices $v_1, v_2, \ldots, v_n$. W.l.o.g. we assume that the $N-n$ extra degree 1 vertices are connected to vertex v_1. The *view* of v_i is what v_i can observe after r rounds of communication. It corresponds to the ball around v_i in P of radius r, including all vertex IDs. Due to the structure of P, if $i \geqslant rt+1$ the view of v_i is entirely determined by the set $B(i) = \{x_{i-rt}, \ldots, x_i, \ldots x_{i+rt}\}$ composed of the IDs of its view. Thus, $B(i)$ is a p-set of $\{1,\ldots,N\}$, for $p = 2rt+1$.

When applied to $P \in \mathscr{C}_t$, algorithm A must decide in particular for each v_i, whether it is selected or not for the dominating set of P. Such a decision can be seen as a 2-coloring c of $V(P)$, where a vertex is assigned color 1 if it is selected by A, and 0 otherwise. Actually, the color $c(v_i)$ depends only on the view of v_i, since the decision of A for v_i is based on the information v_i receives after r rounds of communication. Since for $i \geqslant rt+1$ the view of v_i is entirely determined by the subset $B(i)$, the mapping c restricted to $v_{rt+1}, \ldots, v_{n-rt-1}$ is therefore a mapping from all the p-sets of $\{1, \ldots, N\}$ to $\{0, 1\}$.

Ramsey's Theorem ensures that if $N \geqslant R_2(p, q)$, with $p = 2rt + 1$ and $q = n - 2rt$, there must exist some q-set of IDs taken from $\{1, \ldots, N\}$, say $\{x_{rt+1}, \ldots, x_{n-rt-1}\}$, corresponding to some bad $P \in \mathscr{C}_t$, such that every of its p-sets satisfies $c(B(i)) = c(B(j))$ for all $i, j \in \{rt+1, \ldots, n-rt-1\}$. Note that if this latter range contains enough indices, namely $q \geqslant 2t + 1$, then the three vertices $v_{rt+1}, v_{rt+1+t}, v_{rt+1+2t}$ in this range form a shortest-path of length three forcing one of these vertices at least to be selected in any dominating set. Since $n \geqslant rt + 2t + 1$, we have $q = n - rt \geqslant 2t + 1$, and it follows that $c(B(i)) = c(B(j)) = 1$ for all $i, j \in \{rt+1, \ldots, n-rt-1\}$.

Thus, we showed, for some $P \in \mathscr{C}_t$ with N vertices and with a bad ID distribution taken in $\{1, \ldots, N\}$, that A will select at least $q = n - 2rt$ vertices of P. On the other hand, it is easy to see that in P, $\{*\}$ $v_1, v_{2+2t}, v_{3+4t}, v_{4+6t}, \ldots, v_{s+2(s-1)t}$ form a dominating set of size s for P, where s is the least integer such that $s + 2(s-1)t \geqslant n - t$, i.e., $s = \lceil (n+t)/(2t+1) \rceil = (n+t)/(2t+1)$ by the assumption we made on n (recall that $2t+1$ divides $n+t$). It follows that the approximation ratio α of A is, for every $\varepsilon > 0$,

$$\alpha \geqslant \frac{q}{s} = \frac{n-2rt}{n+t} \cdot (2t+1) = \left(1 - \frac{(2r+1)t}{n+t}\right) \cdot (2t+1) > 2t + 1 - \varepsilon$$

if n is large enough. Note that it is enough to select n such that $(2t+1)(2r+1)t/(n+t) < \varepsilon$, which is equivalent to having $n > (2t+1)(2r+1)t/\varepsilon - t$. □

References

[BCGW21] Marthe Bonamy et al. "A Tight Local Algorithm for the Minimum Dominating Set Problem in Outerplanar Graphs". In: *35th International Symposium on Distributed Computing (DISC)*. Ed. by Seth Gilbert. Vol. 209. LIPIcs. Paris, France, Sept. 2021, 13:1–13:18. DOI: 10.4230/LIPIcs.DISC.2021.13.

[Bod98] Hans Leo Bodlaender. "A partial k-arboretum of graphs with bounded treewidth". In: *Theoretical Computer Science* 209.1-2 (Dec. 1998), pp. 1–45. DOI: 10.1016/S0304-3975(97)00228-4.

[CHW08] Andrzej Czygrinow, Michał Hańćkowiak, and Wojciech Wawrzyniak. "Fast Distributed Approximations in Planar Graphs". In: *22nd International Symposium on Distributed Computing (DISC)*. Vol. 5218. Lecture Notes in Computer Science. Springer, Sept. 2008, pp. 78–92. DOI: 10.1007/978-3-540-87779-0_6.

[CV86] Richard Cole and Uzi Vishkin. "Deterministic Coin Tossing with Applications to Optimal Parallel List Ranking". In: *Information and Control* 70.1 (July 1986), pp. 32–53. DOI: 10.1016/S0019-9958(86)80023-7.

[ES35] Paul Erdős and György Szekeres. "A Combinatorial Problem in Geometry". In: *Composito Mathematica* 2 (1935), pp. 464–470. URL: http://eudml.org/doc/88611.

[GNNW24] Parth Gupta et al. *Optimizing The CGMS Upper Bound on Ramsey Numbers*. Tech. rep. 2407.19026v1 [math.CO]. arXiv, July 2024. DOI: 2407.19026v1.

[HKO+25] Ozan Heydt et al. "Distributed domination on sparse graph classes". In: *European Journal of Combinatorics* 123 (Jan. 2025), p. 103773. DOI: 10.1016/j.ejc.2023.103773.

[HLS14] Miikka Hilke, Christoph Lenzen, and Jukka Suomela. "Brief Announcement: local approximability of minimum dominating set on planar graphs". In: *33rd Annual ACM Symposium on Principles of Distributed Computing* (*PODC*). Paris, France: ACM Press, July 2014, pp. 344–346. DOI: 10.1145/2611462.2611504.

[KSV21] Simeon Kublenz, Sebastian Siebertz, and Alexandre Vigny. "Constant Round Distributed Domination on Graph Classes with Bounded Expansion". In: *28th International Colloquium on Structural Information & Communication Complexity* (*SIROCCO*). Ed. by Tomasz Jurdziński and Stefan Schmid. Vol. 12810. Lecture Notes in Computer Science. Wrocław, Poland: Springer, Cham, June 2021, pp. 334–351. DOI: 10.1007/978-3-030-79527-6_19.

[Lin92] Nathan Linial. "Locality in Distributed Graphs Algorithms". In: *SIAM Journal on Computing* 21.1 (1992), pp. 193–201. DOI: 10.1137/0221015.

[LPW13] Christoph Lenzen, Yvonne-Anne Pignolet, and Roger Wattenhofer. "Distributed minimum dominating set approximations in restricted families of graphs". In: *Distributed Computing* 26.2 (Mar. 2013), pp. 119–137. DOI: 10.1007/s00446-013-0186-z.

[Pan99] Alessandro Panconesi. *Distributed Algorithms Notes* (*manuscript*). 1999.

[RS15] Joel Rybicki and Jukka Suomela. "Exact bounds for distributed graph colouring". In: *22nd International Colloquium on Structural Information & Communication Complexity* (*SIROCCO*). Vol. 9439. Lecture Notes in Computer Science. Montserrat, Spain: Springer, July 2015, pp. 46–60. DOI: 10.1007/978-3-319-25258-2_4.

Part C

Pathwidth to the rescue

Chapter VI
An introduction to pathwidth

VI.1 Pathwidth

Pathwidth measures how path-like and linear a graph is. It was introduced by Robertson and Seymour in the first paper of their celebrated *Graph Minors* series [RS83].

Path decomposition. A *path decomposition* of a graph G is a sequence $\mathcal{P} = (X_1, X_2, \ldots, X_q)$ of vertex subsets of G, called *bags*, such that

- $V(G) = \bigcup_{i=1}^{q} X_i$,
- for every edge $\{x, y\} \in E(G)$ there is at least one bag containing both endpoints, and
- for every vertex $x \in V(G)$, the bags containing x form a contiguous subsequence of $\mathcal{P}$.

The *width* of $\mathcal{P}$ is $\max\{|X_i| - 1 \mid 1 \leqslant i \leqslant q\}$, and the *pathwidth* $\mathrm{pw}(G)$ of G is the smallest width over all path decompositions of G. The bags X_1 and X_q are called the *extremities* of $\mathcal{P}$. We refer to X_1 as the *leftmost bag* and to X_q as the *rightmost bag*.

An equivalent and convenient reformulation of the contiguity is that for all indices $i \leqslant j \leqslant k$, one has $X_i \cap X_k \subseteq X_j$.

Example. Consider the path P_4 on vertices v_1, v_2, v_3, v_4 with edges v_1v_2, v_2v_3, v_3v_4. The sequence

$$X_1 = \{v_1, v_2\}, \quad X_2 = \{v_2, v_3\}, \quad X_3 = \{v_3, v_4\}$$

is a valid path decomposition: every edge appears in some bag, and each vertex appears in a contiguous block of bags (see Section VI.1). The largest bag has size 2, giving width 1, i.e. $\mathrm{pw}(P_4) \leqslant 1$.

Graphs of low pathwidth. A connected graph has pathwidth 1 if and only if it is a *caterpillar*, that is, a tree in which all vertices are neighbors of a central path. To construct a path-decomposition of a caterpillar, one can order all edges from left to right and go through edges in order one at a

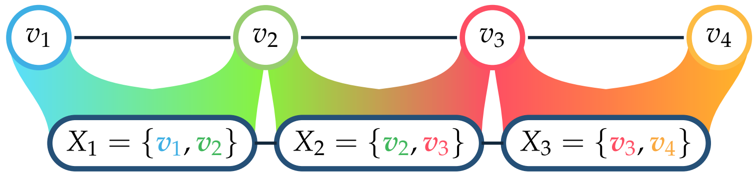


Figure VI.1: A path decomposition of P_4 of width 1.

time by putting each one in a bag. To see that pathwidth 1 graphs are caterpillar, one can observe that, intuitively, with bags of size at most 2, we can only carry one edge's worth of interface at a time, which forces a single path with pendant leaves.

For $n \geqslant 3$, the cycle C_n has $\mathrm{pw}(C_n) = 2$. A lower bound of 2 follows from the fact that C_n is not a caterpillar. For the upper bound, one can use the following folklore result.

Remark VI.1.1. *For every graph G and $X \subseteq V(G)$, we have the following:*

$$\mathrm{pw}(G) \leqslant |X| + \mathrm{pw}(G - X).$$

As a cycle minus a vertex is a path (of pathwidth 1), this immediately implies an upper bound of 2. In more detail, one can take bags of the form $\{v_1, v_i, v_{i+1}\}$ for $i = 1, \ldots, n-1$ (see Figure VI.2). This follows the intuition that low pathwidth graph are path-like.

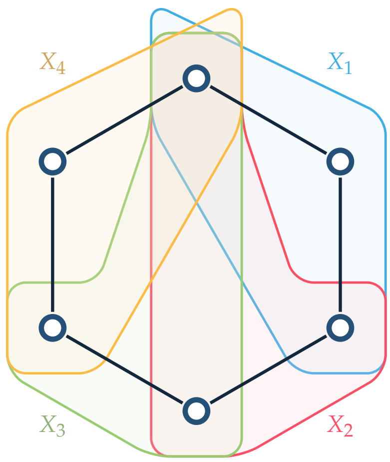


Figure VI.2: A path decomposition of C_6.

A key observation is that the intersection of two bags (called an *adhesion*) always separate the union of the bags at the left from the union of the bags at the right. More formally, we have the following folklore result (see also Section VI.1).

Proposition VI.1.2. *Let $\mathcal{P} = (X_1, X_2, \ldots, X_q)$ be a path decomposition of G, and let $1 \leqslant i < q$. Then the adhesion $X_i \cap X_{i+1}$ separates $\bigcup_{j \leqslant i} X_j$ from $\bigcup_{j>i} X_j$.*

Therefore, every $X_i \cap X_{i+1}$ serves as an interface between what happens on the left, and what happens on the right.

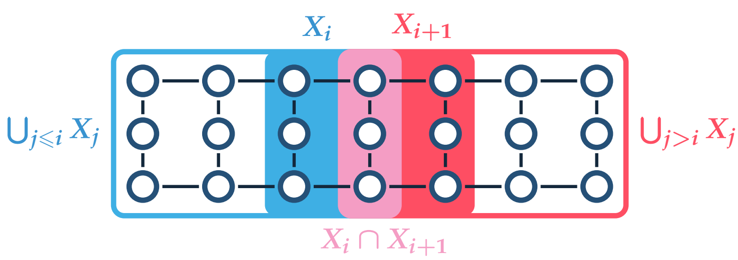


Figure VI.3: An illustration of Proposition VI.1.2.

VI.2 High pathwidth, and lower bounds

Graphs of high pathwidth. Regular trees are a simple example of non path-like graph. Non surprisingly, they have large pathwidth. Indeed, let T_h denote the complete binary tree of height h. One can show by a classical argument [KS93] that $\mathrm{pw}(T_h) = \lceil h/2 \rceil$. In particular, trees on n vertices can have pathwidth $\Omega(\log n)$. Thus, pathwidth is unbounded on the class of trees.

Interestingly, there is sort of a converse to this. The Graph Minor theory characterizes bounded pathwidth classes via excluded forest minors: a class $\mathcal{C}$ of graphs has bounded pathwidth if and only if some fixed forest F is not a minor of any graph in $\mathcal{C}$ [RS83]. Although the bounds obtained this way are not always tight, this is a great way to show that a class of graphs has bounded pathwidth (by showing some fixed forest is excluded as a minor). As pathwidth is a minor-closed property, one can also show that the pathwidth of a class of graphs is unbounded by exhibiting graphs of the form T_h for unbounded h.

Another simple (but important) example of high pathwidth graphs are cliques. One can show that every clique of G must be contained in some single bag of any path decomposition. This follows immediately from the contiguity condition: if $K \subseteq V(G)$ is a clique, then for any two vertices $u, v \in K$, there is a bag containing both. By the Helly property on intervals, all bags containing vertices of K have a common index, so all of K is contained in one of the bags. Consequently, $\mathrm{pw}(G) \geqslant \omega(G) - 1$, where $\omega(G)$ denotes the clique number of G. For the complete graph K_n, this bound is tight: $\mathrm{pw}(K_n) = n - 1$, witnessed by path-decomposition with a single bag with vertex set $V(K_n)$.

Lower bound via node searching. Pathwidth admits a beautiful combinatorial characterization in terms of *node searching*. We will reuse in what follows some of the vocabulary from [DKT97]. The *node searching game* [KP85] is a pursuit-evasion game on G. Intuitively, in this game, k cops try to catch an invisible robber moving on the vertices of G. By invisible, we mean that before taking a decision, the robber already knows the whole sequence of moves that the cops will follow in the future, and can adapt its own strategy with respect to this knowledge. More precisely, a *search with k cops* on G is a finite sequence $C_1, \ldots, C_t$ of subsets of vertices of G of size at most k such that, for $i = 2, \ldots, t$, the symmetric difference between C_i and C_{i-1} has size at most 1. Each set C_i represents vertices occupied by k or fewer cops, and at each step, a cop is either added or removed (or nothing changed). Given such a search on G, we define inductively its corresponding *free locations* $F_1, \ldots, F_t$ by setting $F_1 := V(G) \setminus C_1$ and for each $i \in \{2, \ldots, t\}$, by letting F_i be the set of all vertices v from $V(G) \setminus C_i$ for which there exists some path from some vertex $u \in F_{i-1}$ to v, with all vertices except possibly u in $V(G) \setminus C_i$. Informally, F_i represents the set of all vertices that could be reached by the robber at step i, while evading the cops so far. We say that a search $C_1, \ldots, C_t$ is *evasive* if the robber can escape the cops with respect to this search, i.e. if $C_t \neq \varnothing$. The *node searching number* of G, denoted $\mathrm{ns}(G)$ is the minimum possible integer $k \geqslant 0$ for which G admits a non-evasive search with k cops. Intuitively, $\mathrm{ns}(G) > k$ means that when playing with k cops, whatever strategy the cops choose, the robber will be able to find accordingly a strategy to escape them. A classic result of Kirousis and Papadimitriou [KP85] states that for every graph G, we have $\mathrm{ns}(G) = \mathrm{pw}(G) + 1$. This gives an intuitive view of pathwidth: it is the number of vertices that need to be guarded at each time step, in order to sweep through G without letting the fugitive escape back into already-cleared territory. This point of view is also very useful to prove pathwidth lower bounds (and is used in this manuscript in Section VII.6). Indeed, to prove a lower bound on $\mathrm{ns}(G)$, one just has to describe an evasive strategy for the robber.

References

[DKT97] Nick D Dendris, Lefteris M Kirousis, and Dimitrios M Thilikos. “Fugitive-search games on graphs and related parameters”. In: *Theoretical Computer Science* 172.1-2 (1997), pp. 233–254.

[KP85] Lefteris M. Kirousis and Christos H. Papadimitriou. “Interval graphs and searching”. In: *Discrete Mathematics* 55.2 (1985), pp. 181–184. ISSN: 0012-365X. DOI: https://doi.org/10.1016/0012-365X(85)90046-9. URL: https://www.sciencedirect.com/science/article/pii/0012365X85900469.

[KS93] Ephraim Korach and Nir Solel. “Tree-width, path-width, and cutwidth”. In: *Discrete Applied Mathematics* 43.1 (1993), pp. 97–101. ISSN: 0166-218X. DOI: https://doi.org/10.1016/0166-218X(93)90171-J. URL: https://www.sciencedirect.com/science/article/pii/0166218X9390171J.

[RS83] Neil Robertson and P.D. Seymour. “Graph minors. I. Excluding a forest”. In: *Journal of Combinatorial Theory, Series B* 35.1 (1983), pp. 39–61. ISSN: 0095-8956. DOI: https://doi.org/10.1016/0095-8956(83)90079-5. URL: https://www.sciencedirect.com/science/article/pii/0095895683900795.

Chapter VII

A polynomial bound on the pathwidth of graphs edge-coverable by k shortest paths

Abstract

Dumas, Foucaud, Perez and Todinca (2024) recently proved that every graph whose edges can be covered by k shortest paths has pathwidth at most $O(3^k)$. In this paper, we improve this upper bound on the pathwidth to a polynomial one; namely, we show that every graph whose edge set can be covered by k shortest paths has pathwidth $O(k^4)$, answering a question from the same paper. Moreover, we prove that when $k \leqslant 3$, every such graph has pathwidth at most k (and this bound is tight). Finally, we show that even though there exist graphs with arbitrarily large treewidth whose vertex set can be covered by 2 isometric trees, every graph whose set of edges can be covered by 2 isometric trees has treewidth at most 2.

Acknowledgements

This chapter presents a joint work with Julien Baste, Lucas De Meyer, Ugo Giocanti and Etienne Objois published at STACS'26 [BMG+26]. We would like to thank the anonymous reviewers for their instructive feedbacks about the conference version of this paper. In particular one of them pointed out to us that the lower bound from Proposition VII.1.3 did not follow from [DFPT24] as we had claimed in a previous version of the paper. We thank Marcin Briański for the example from Section VII.1 (see Figure VII.1) of graphs vertex-coverable by two isometric trees with large treewidth. We also thank Florent Foucaud for mentioning the potential algorithmic application of Theorem VII.1.2 to Isometric Path Edge-Cover.

Contents

VII.1 Introduction

Graph covering is an old and recurrent topic in graph theory. For a fixed class $\mathcal{H}$ of graphs, the covering literature aims to answer two types of problems. First are optimization problems: given a graph G, what is the minimum number of graphs from $\mathcal{H}$ needed to "cover"[1] G? We refer to Knauer and Ueckerdt [KU16] for a more general overview on the topic. The second type are structural problems: for the fixed class $\mathcal{H}$ of graphs and some $k \geqslant 1$, what is the structure of the graphs that can be "covered" by (at most) k graphs from $\mathcal{H}$?

[1] Various notions of coverings can usually be considered here.

We say that a graph G is *edge-coverable* (resp. *vertex-coverable*) by some subgraphs $H_1, \ldots, H_k$ if every *edge* (resp. *vertex*) of G belongs to at least one of the subgraphs H_i. Motivated by some algorithmic applications to the Isometric Path Cover problem and its variants, Dumas, Foucaud, Perez and Todinca [DFPT24] recently worked on a particular metric version of such a covering problem. They proved that every graph edge-coverable (resp. vertex-coverable) by few shortest paths shares structural similarities with a simple path. More precisely, if a graph is edge-coverable or vertex-coverable by k shortest paths, its *pathwidth* is bounded by an exponential in k (see Section VII.2 for a definition of pathwidth).

Theorem VII.1.1 ([DFPT24])**.** *Let $k \geqslant 1$ and G be a graph vertex-coverable by k shortest paths. Then $\mathrm{pw}(G) = O(k \cdot 3^k)$. Moreover, if G is edge-coverable by k shortest paths, then $\mathrm{pw}(G) = O(3^k)$.*

The proof of Theorem VII.1.1 uses a branching approach to show that, in every graph vertex-coverable (resp. edge-coverable) by at most k shortest paths, and for every vertex u and every $i \geqslant 1$, the number of vertices at distance exactly i from u in G is $O(k \cdot 3^k)$ (resp. $O(3^k)$). We also mention that a "coarse" variant of Theorem VII.1.1 has recently been proved [HP25]. Then, the bags of the path decomposition proving Theorem VII.1.1 are the union of every two consecutive layers in a BFS-layering of G (starting from an arbitrary vertex). Such a path decomposition is clearly not optimal in general: if we let S_h denote the star with h leaves, then $\mathrm{pw}(S_h) = 1$, while such a decomposition will always have width at least $\lfloor \frac{h}{2} \rfloor$, whatever the choice of u is. Dumas et al. [DFPT24] asked if the bounds from Theorem VII.1.1 can be made polynomial in k. Our main result is a positive answer in the edge-coverable case.

Theorem VII.1.2. *Let $k \geqslant 1$, and G be a graph edge-coverable by k shortest paths. Then $\mathrm{pw}(G) = O(k^4)$.*

Our proof uses a different approach from the one of [DFPT24]. Roughly speaking, it consists in finding some separator[2] X of polynomial size, such that for every component C of $G - X$, C has a "nice" layering with

[2] meaning, some $|X|$-cut

respect to one of the k covering paths.

In terms of lower bounds, we provide for every $k \geqslant 1$, a construction of a graph edge-coverable by k shortest paths and with pathwidth equal to k.

Proposition VII.1.3. *For any integer $k \geqslant 1$, there is a graph G_k that is*

(a) *edge-coverable by k shortest paths, and*

(b) *that has pathwidth at least k.*

To our knowledge, no better lower bound is known. Our second main result is that this lower bound is tight for $k \leqslant 3$.

Theorem VII.1.4. *Let $k \leqslant 3$, and G be a graph edge-coverable by k shortest paths. Then* $\mathrm{pw}(G) \leqslant k$.

Given a graph G, a subgraph H of G is *isometric* if for every two vertices x, y of H, the distances between x and y are the same in G and H. In other words, if x and y are connected in G, then there must exist a shortest path between them in G which is included in H.

In view of the previous results, the following question naturally arises: given a family $\mathcal{H}$ of graphs, what is the structure of a graph G edge/vertex-coverable by at most k *isometric* subgraphs all isomorphic to graphs in $\mathcal{H}$? For instance, if $\mathcal{H}$ is the family of all trees, does G somehow look like a tree?

Extending a result of Aigner and Fromme [AF84] for paths, Ball, Bell, Guzman, Hanson-Colvin and Schonsheck [BBG+17] proved that every graph vertex-coverable by k isometric subtrees has *cop number* at most k. In particular, it is well-known that the cop number of a graph is always upper-bounded by its *treewidth* (see Section VII.5 for a definition of treewidth). In analogy to Theorem VII.1.1, a natural question is then the following: is it true that every graph which is vertex/edge-coverable by a small number of isometric subtrees has small treewidth? This question turns out to have a negative answer when considering vertex-coverability, as shown by the following example, pointed out to us by Marcin Briański (personal communication): for every $t \geqslant 1$, let G_t be the graph on $2t + 2$ vertices obtained from the biclique $K_{t,t}$ after adding two vertices a, b of degree t respectively adjacent to all vertices from each of the sides of $K_{t,t}$ (see Figure VII.1 for the case $t = 3$).

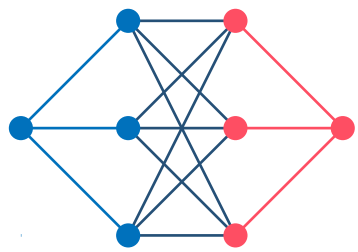


Figure VII.1: This graph is vertex-coverable by 2 isometric trees (one with red vertices, the other with blue vertices) and contains $K_{3,3}$ as a minor (black edges). Hence, it has treewidth at least 3.

Then, G_t is vertex-coverable by the two induced stars S_a, S_b respectively centered in a and b (which are in particular isometric subgraphs of G_t), but has treewidth at least t, as it contains $K_{t,t}$ as a minor. Our third result implies that any such construction fails when considering graphs which are edge-coverable by two isometric trees, as we show that they have treewidth at most 2.

Theorem VII.1.5. *Let G be a graph which is edge-coverable by 2 isometric trees. Then* $\mathrm{tw}(G) \leqslant 2$.

Algorithmic motivations. *Routing problems* play a central role not only in algorithmic graph theory and complexity, but also in structural graph theory. One of the best examples is the Disjoint Paths problem, appearing in the seminal Graph Minor series of papers of Robertson and Seymour [RS95], in which one is given as an input a graph G, an integer k and k pairs of vertices, and one must decide whether there exists k pairwise disjoint paths connecting all pairs of vertices. This problem has been shown to be solvable in FPT time, when parameterized by the number k of shortest paths [RS95, KKR12], and this result has been a key ingredient to design efficient algorithms deciding the existence of *minors* in graphs. The metric variant Disjoint Shortest Path of this problem [Eil98] (in which one further wants to decide whether there exist k pairwise disjoint shortest paths connecting all input pairs of vertices) has attracted significant attention, and was recently proved to be solvable in XP time, when parameterized by k [BK17, Loc21, BNRZ23].[3]

[3] It is worth mentioning that all the aforementioned problems are NP-hard [FHW80, Loc21].

A number of metric variants of routing problems of similar flavor has been studied over the last decades. Among them, Isometric Path Cover [AF84, FF01] takes as input a graph G and an integer k, and asks whether G is vertex-coverable by k shortest paths. Its routing version Isometric Path Cover With Terminals takes as input a graph G, some integer k, and k pairs of vertices, and asks whether there exist k shortest paths that vertex-cover G and connect all the input pairs of vertices. Both problems are NP-complete [CDD+22, DFPT24]. By combining Theorem VII.1.1 with Courcelle's theorem [Cou90], it was shown [DFPT24] that there exists algorithms running in XP (respectively FPT time) when parameterized by k, to solve Isometric Path Cover (respectively Isometric Path Cover With Terminals). In particular, it is still open [DFPT24] whether the Isometric Path Cover problem admits an FPT algorithm parameterized by k. We also refer to Chakraborty et al. [CDF+25] for some recent work on similar metric versions of routing problems. It is worth mentioning that Isometric Path Cover admits some polynomial time c-approximation algorithms on graphs with *isometric path complexity* at most c [CCFV26], a recent metric parameter of graphs [CDD+22] (introduced under the name of *isometric path antichain cover number*).

As mentioned above, the original parameterized algorithms solving Isometric Path Cover and its variant with terminals [DFPT24] rely on Courcelle's theorem, and thus the dependencies in k are highly non-trivial and do not give rise in general to practical algorithms. Chakraborty et al. [CDF+25, Section 3] designed a dynamic programming procedure solving the partition variant Isometric Path Partition of Isometric Path Cover in time $n^{O(\mathrm{tw}(G))}$, where $\mathrm{tw}(G)$ denotes the treewidth of the input graph. As pointed out to us by Florent Foucaud (personal communication), it is very likely that combining our bound from Theorem VII.1.2 with such an approach could lead to better running times than the ones given by Courcelle's theorem for algorithms solving Isometric Path Edge-Cover and its version with terminals (the natural edge-covering variant of Isometric Path Cover).

Organization of the paper. Section VII.2 contains all preliminary definitions and some basic results related to pathwidth. Section VII.3, Section VII.4, Section VII.5 and Section VII.6 contain respectively proofs of Theorems VII.1.2, VII.1.4 and VII.1.5 and Proposition VII.1.3. Note that these sections are completely independent of each other, and can be read in any order. Section VII.7 contains some discussion and additional questions left open by this work.

VII.2 Preliminaries

VII.2.1 Notations and Basic Definitions

Graphs. For all subsets $X, Y \subseteq V(G)$ of vertices, we denote $E_G(X,Y) := \{xy \in E(G) \mid x \in X \setminus Y, y \in Y \setminus X\}$. If $U \subseteq V(G)$, we let $N_{G,U}(X)$ denote the set of vertices in $U \setminus X$ having at least one neighbor in X. When the context is clear, we will simply write $N_U(X)$.

Paths. For each $i \leqslant j$, we let $P[v_i, v_j]$ denote the subpath $v_i, v_{i+1}, \ldots, v_j$ of P, and we let P^{-1} denote the path $v_k, \ldots, v_1$. For every two paths $P = v_1, \ldots, v_p$ and $Q = w_1, \ldots, w_q$ such that $v_p = w_1$, and such that $P - v_p$ is disjoint from $Q - w_1$, we define the path $P \cdot Q := v_1, \ldots, v_p = w_1, \ldots, w_q$. If $P = v_1, \ldots, v_k$ is a path, for every $1 \leqslant i < j \leqslant k$, we say that v_i is *before* v_j on P, and that v_j is *after* v_i on P. A path P is *vertical* with respect to a vertex u, if for every $i \in \mathbb{N}$, P has at most one vertex at distance i from u in G. Note that every path vertical with respect to some vertex u is a shortest path, and that every shortest path starting from u is vertical with respect to u.

Covers. Let G be a graph and $H_1, \ldots, H_k$ be subgraphs of G. We say that the graphs $H_1, \ldots, H_k$ *edge-cover* G if $E(G) = \bigcup_{i=1}^{k} E(H_i)$. Similarly, we say that $H_1, \ldots, H_k$ *vertex-cover* G if $V(G) = \bigcup_{i=1}^{k} V(H_i)$. Note that whenever G does not contain an isolated vertex, then every edge-cover is also a vertex-cover.

Metrics. We let $\mathrm{d}_G(\cdot,\cdot)$ (or simply $d(\cdot,\cdot)$ when the context is clear) denote the shortest-path metric of a graph G. Recall that a graph H is an *isometric subgraph* of G if H is a subgraph of G such that for every two vertices $x, y \in V(H)$, we have $d_H(x,y) = d_G(x,y)$. For every vertex v of a graph G, we let $\mathrm{ecc}(v) := \max\{d_G(u,v) \mid u \in V(G)\} \in \mathbb{N} \cup \{\infty\}$ denote the *eccentricity* of v.

Orders. Let $(X, \leqslant)$ be a partially ordered set. For every two disjoint subsets $A, B \subseteq X$, we write $A < B$ if $a < b$ for every $(a,b) \in A \times B$. Moreover, for every $x \in X$ and every $A \subseteq X$, we let $A|_{\leqslant x} := \{a \in A \mid$

$a \leqslant x\}$ and $A|_{\geqslant x} := \{a \in A \mid a \geqslant x\}$. We define similarly $A|_{<x}$ and $A|_{>x}$. A *chain* is a sequence $(x_1, \ldots, x_k)$ of distinct elements of X such that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_k$.

VII.2.2 Pathwidth and layerings

We collect in this section a number of basic properties of pathwidth.

Vertex-separation number. Let G be a graph and $<$ denote a total ordering on $V(G)$. The vertex-separation number of $(G, <)$ is the value $\min_{w \in V(G)} |\{u \in V(G) \mid u < w \text{ and } \exists v \geqslant w, uv \in E(G)\}|$. The vertex separation number of G is the minimum over the vertex separation numbers of $(G, <)$, for all possible linear orderings $<$ of G.

Theorem VII.2.1 ([Kin92]). *The vertex separation number of a graph equals its pathwidth.*

Layerings. A *layering* of a graph G is a partition of its vertices into a sequence $(V_1, \ldots, V_t)$ of subsets of $V(G)$ called *layers*, such that each graph $G[V_i]$ is an independent set, and for every $i, j \in \{1, \ldots, t\}$ such that $|j - i| \geqslant 2$, we have $E_G(V_i, V_j) = \varnothing$.[4]

A *k-layering* of G is a layering $(V_1, \ldots, V_t)$ such that for every $0 < i < t$, we have $|E_G(V_i, V_{i+1})| \leqslant k$. Note that in particular, it implies that $|N_{V_{i+1}}(V_i)| \leqslant k$ and $|N_{V_i}(V_{i+1})| \leqslant k$. The bags V_1 and V_t are the *extremities* of the layering.

[4] Note that our definition of layering differs from the one which is commonly used in the literature, as one usually allows edges between vertices from a same layer.

Lemma VII.2.2. *Let G be a graph admitting a k-layering. Then G has pathwidth at most k. Moreover, if G is connected, then G has a path decomposition of width at most k, whose extremities are exactly the extremities of the k-layering.*

Proof. We let $(V_1, \ldots, V_t)$ denote a k-layering of G, and assume without loss of generality that G has no isolated vertex. We also set $V_0 := V_{t+1} := \varnothing$. For every $i \in \{1, \ldots, t\}$, we set $V_i^\ell := N_{V_i}(V_{i-1})$ and $V_i^r := N_{V_i}(V_{i+1})$. Note in particular that, as G has no isolated vertex, for each $i \in \{1, \ldots, t\}$ we have $V_i = V_i^\ell \cup V_i^r$, $|V_i^\ell| \leqslant k$, $|V_i^r| \leqslant k$, and $V_1^\ell = V_t^r = \varnothing$, and that for each $i < t$, all the edges of G between V_i and V_{i+1} are between V_i^r and V_{i+1}^ℓ.

We will define a total order $\leqslant$ on $V(G)$ such that $(G, \leqslant)$ has vertex separation number at most k. In particular, Theorem VII.2.1 will then immediately imply that $\mathrm{pw}(G) \leqslant k$. We let $\leqslant$ be any total ordering on $V(G)$ satisfying the following two properties:

- for every $i < j$, we have $V_i < V_j$;
- for every $i \in \{1, \ldots, t\}$, we have $V_i^\ell \setminus V_i^r < V_i^\ell \cap V_i^r < V_i^r \setminus V_i^\ell$.

We now show that $(G, \leqslant)$ has vertex separation number at most k and let $w \in V(G)$. Set $S_w := \{u : u < w \text{ and } \exists v \geqslant w \mid uv \in E(G)\}$. Our goal is to show that $|S_w| \leqslant k$. We let i denote the unique element from $\{1,\dots,t\}$ such that $w \in V_i$. We define the sets $A := V_i^r|_{<w}$, $B := V_i^\ell|_{\geqslant w}$ and $C := V_{i-1}^r$. Note that A, B, C are pairwise disjoint, and that moreover, as $(V_1,\dots,V_t)$ is a layering of G, $A \subseteq S_w \subseteq A \sqcup C$.

First, observe that if B is empty, then we must have $S_w \cap C = \varnothing$, so that $S_w = A$, and thus as every vertex of A is by definition of V_i^r adjacent to a vertex of V_{i+1}, and as $|E_G(V_i, V_{i+1})| \leqslant k$, we then have $|S_w| \leqslant k$. Assume now that B is not empty. Then there exists an element $v \geqslant w$ such that $v \in V_i^\ell$. Thus, by definition of $<$, we have $V_i|_{\leqslant w} \subseteq V_i^\ell$, and thus $A \subseteq V_i^\ell$.

We now set $k_1 := |E_G(C, A)|$ and $k_2 := |E_G(C, B)|$. As $A \subseteq V_i^\ell$, every element of A must be incident to an edge from $E_G(A, C)$, hence $|A| \leqslant k_1$. As $C = V_{i-1}^r$, we have $|C| \leqslant k_2$. Moreover, note as $(V_1,\dots,V_t)$ is a k-layering and as A and B are disjoint, we have $k_1 + k_2 \leqslant k$. In particular, it implies that

$$|S_w| \leqslant |A| + |C| \leqslant k_1 + k_2 \leqslant k,$$

hence we can conclude that $(G, \leqslant)$ has indeed vertex separation number at most k.

To prove the "Moreover" part of the lemma, observe that if G is connected, then both V_1 and V_t have size at most k. We claim that it is not hard to check that the path decomposition given by Theorem VII.2.1 can then be slightly modified so that its extremities are V_1 and V_k. $\square$

VII.3 A General Polynomial Bound

In this section, we prove Theorem VII.1.2, which we restate here.

Theorem VII.1.2. *Let $k \geqslant 1$, and G be a graph edge-coverable by k shortest paths. Then $\mathrm{pw}(G) = O(k^4)$.*

Throughout this section, we let G be a graph whose edge set can be covered by a set $\mathcal{P} := \{P_1,\dots,P_k\}$ of k isometric paths. Our main result is the following, which will immediately imply a proof of Theorem VII.1.2.

Theorem VII.3.1. *For every $i \in \{1,\dots,k\}$, there exists a set of vertices $X_i \subseteq V$ of size at most $720k^3 + 4k$ such that every connected component of $G - X_i$ that intersects P_i has pathwidth at most $6k$.*

Proof that Theorem VII.3.1 implies Theorem VII.1.2. For every $1 \leqslant i \leqslant k$, we use Theorem VII.3.1 to obtain the set X_i. We consider their union $X := \bigcup_{i=1}^k X_i$. Observe that $|X| \leqslant \sum_{i\in[k]} |X_i| \leqslant k \cdot (720k^3 + 4k) = 720k^4 + 4k^2$. We now let C be a connected component of $G - X$. As $P_1,\dots,P_k$ edge-cover (and thus also vertex-cover) G, there is some $i \in \{1,\dots,k\}$ such that the path P_i intersects C. In particular, as C is included in a connected component of $G - X_i$, it implies that C has pathwidth at most $6k$, and thus that $\mathrm{pw}(G - X) \leqslant 6k$. Using Remark VI.1.1 we conclude that $\mathrm{pw}(G) \leqslant |X| + \mathrm{pw}(G - X) \leqslant 720k^4 + 4k^2 + 6k$. □

The remainder of the section will consist in a proof of Theorem VII.3.1. In Section VII.3.1 we prove some preliminary results, and show that our proof reduces to the problem of finding a special kind of separations in a subgraph G_0 of G which is $6k$-layered (see Lemma VII.3.9). In Section VII.3.2, we give a proof of Lemma VII.3.9, which is the most technical part of our proof.

Without loss of generality, we will assume from now on that $i = 1$, i.e., we will construct a set X of size at most $720k^3 + 4k$ such that every component of $G - X$ that intersects P_1 has pathwidth at most $6k$.

VII.3.1 Preliminary Results

Parallel paths. Let P be a shortest path between two vertices a, b of G. A path Q of G is *parallel* to P in G if Q is a subpath of a shortest ab-path of G.

Note that this does not define a symmetric relation in general: the fact that a path Q is parallel to another path P does not necessarily imply that P is parallel to Q. However, in the remainder of the proof, we will always consider the property of being parallel to the path P_1, which is fixed and connects vertices a_1 and b_1. In particular, the following simple observation implies that every graph which is edge-coverable by a few number of paths that are all parallel to P_1 has a very restrained structure.

Lemma VII.3.2. *Let $\{Q_1, \ldots, Q_\ell\}$ be a collection of paths in G that are all parallel to P_1. Then the graph $\bigcup_{j=1}^{\ell} Q_j$ is ℓ-layered.*

Proof. Let $U := \bigcup_{i=1}^{\ell} V(Q_i)$, let $r := d(a_1, b_1)$, and for every $i = 0, \ldots, r$, let $U_i := \{u \in U \mid d(a_1, u) = i\}$. As every path Q_j is parallel to P_1, observe that for each $(i, j) \in \{0, \ldots, r\} \times \{1, \ldots, \ell\}$, we must have $|V(Q_j) \cap U_i| \leqslant 1$. In particular, for each $i \in \{0, \ldots, r\}$, we have $|U_i| \leqslant \ell$. Hence, $(U_0, \ldots, U_r)$ forms a partition of U, such that each edge of $G[U]$ has its two endvertices in some pair $(U_i, U_{i'})$ such that $|i' - i| \leqslant 1$. Moreover, as the paths Q_j are all parallel to P_1, no edge of a path Q_j can have both endvertices in the same set U_i, so $(U_0, \ldots, U_r)$ is indeed an ℓ-layering of $\bigcup_{j=1}^{\ell} Q_j$. □

Lemma VII.3.3. *If P is a path parallel to P_1 with endvertices $u, v \in V(G)$, then every shortest uv-path in G is also parallel to P_1.*

Proof. This immediately follows from the observation that every subpath of a shortest path is also a shortest path. □

We consider the partial order $\leqslant_1$ on $V(G)$ defined by setting for every two vertices $u, v \in V(G)$, $u \leqslant_1 v$ if and only if $d_G(a_1, u) \leqslant d_G(a_1, v)$. Note in particular that the vertex set of every path P which is parallel to P_1, forms a chain with respect to $\leqslant_1$.

Lemma VII.3.4. *Let $a \leqslant_1 b \leqslant_1 c$ be three vertices such that there exist two paths P, Q both parallel to P_1, such that P connects a to b and Q connects b to c. Then $P \cdot Q$ is parallel to P_1.*

Proof. Note that $P \cdot Q$ must be a shortest ac-path, as for each $i \geqslant 0$, it contains at most one vertex at distance exactly i from a_1. In particular, it is not hard to conclude that $P \cdot Q$ is parallel to P_1. □

Reducing $\mathcal{P}$. The first step in the proof of Theorem VII.3.1 consists in modifying the covering family $\mathcal{P}$ in order to make it reduced (see definition below). The intuition behind this step is that our proof of Theorem VII.3.1 works particularly well when all paths in $\mathcal{P}$ intersect P_1. Informally, the reduction operation consists in making $\mathcal{P}$ contain as many paths that intersect P_1 as possible. This step will turn out to be useful at the very end of our proof (see the proof of Lemma VII.3.11).

A shortest path Q of G is called a *reducing path* if:

- Q is disjoint from P_1, and admits a subpath P parallel to P_1, that connects two vertices u, v, such that $u <_1 v$,
- there exists a shortest path R in G from u to v that intersects P_1.

Section VII.3.1 depicts a reducing path. If a family $\mathcal{Q}$ of shortest paths has no reducing path, then we say that $\mathcal{Q}$ is *reduced*.

Proposition VII.3.5. *Let $\mathcal{Q}$ be a family of shortest paths. Then there exists a family $\mathcal{Q}'$ of shortest paths which is reduced, such that $|\mathcal{Q}'| \leqslant 2|\mathcal{Q}|$ and that the paths of $\mathcal{Q}'$ cover all the edges covered by the paths of $\mathcal{Q}$, i.e., such that*

$$\bigcup_{P \in \mathcal{Q}} E(P) \subseteq \bigcup_{P \in \mathcal{Q}'} E(P).$$

Proof. Assume that $\mathcal{Q}$ is not reduced, and let Q be a reducing path in $\mathcal{Q}$. We also let P be a subpath of Q with endvertices u, v such that $u <_1 v$, and R be a shortest path in G from u to v intersecting P_1. Let R_1, R_2 denote two shortest paths respectively from a_1 to u and from v to b_1. We also write $Q = Q_1 \cdot Q_2 \cdot Q_3$, so that Q_2 is the uv-subpath of Q parallel to P_1 (see Section VII.3.1). Observe now that both paths $Q' := Q_1 \cdot R \cdot Q_3$ and $Q'' := R_1 \cdot Q_2 \cdot R_2$ are shortest paths in G that both intersect P_1 and cover all the edges of Q. In particular, it implies that the family of paths $\mathcal{Q}' := (\mathcal{Q} \setminus \{Q\}) \cup \{Q', Q''\}$ covers all the edges covered by the paths of $\mathcal{Q}$, and has strictly fewer paths than $\mathcal{Q}$ that do not intersect P_1. We apply iteratively the same operation as long as the obtained family $\mathcal{Q}'$ is not reduced. In particular, as at each step, the number of paths of $\mathcal{Q}$ which are disjoint with P_1 strictly decreases, then after at most $|\mathcal{Q}|$ iterations, we obtain a reduced family with the desired properties. □

Up to applying Proposition VII.3.5 to $\mathcal{P}$, we will assume from now on that $\mathcal{P}$ is a reduced family of at most $2k$ shortest paths of G that edge-cover G.

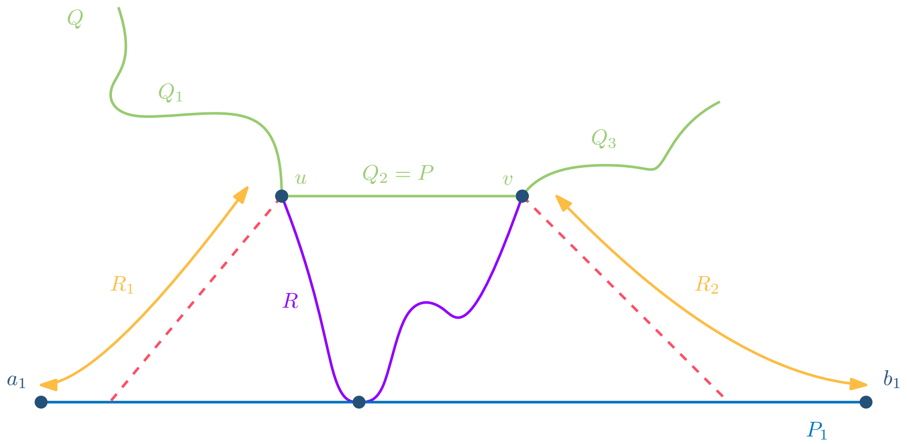


Figure VII.2: The path Q represented in green is reducing. P_1 is represented in blue.

Good and bad vertices. We now partition the set $\mathcal{P} := \{P_i \mid 1 \leqslant i \leqslant 2k\}$ into the subset $\mathcal{P}_1$ of the paths in $\mathcal{P}$ that intersect P_1, and $\mathcal{P}_2 := \mathcal{P} \setminus \mathcal{P}_1$.

For every $P \in \mathcal{P}_1$, we let C_P denote the maximal subpath of P with both endvertices a_P, b_P in $V(P_1)$, so that $a_P \leqslant_1 b_P$ (up to considering P^{-1} instead of P, we also assume that a_P is before b_P on P). Note that $C_{P_1} = P_1$, and that by definition of $\mathcal{P}_1$, each C_P has at least one vertex, and is parallel to P_1. Moreover, consider for every $P \in \mathcal{P}_1$ the two vertices u_P, v_P that respectively maximize $d_G(u_P, a_P)$ and $d_G(b_P, v_P)$, such that u_P, a_P, b_P, v_P appear in this order on P, and such that the paths $P[u_P, a_P]$ and $P[b_P, v_P]$ are parallel to P_1. We now let $\widetilde{C}_P := P[u_P, v_P]$ and A_P, B_P denote the two subpaths of P such that $P = A_P \cdot \widetilde{C}_P \cdot B_P$ (see Section VII.3.1). We stress out that in general, the path $\widetilde{C}_P$ is not necessarily parallel to P_1.

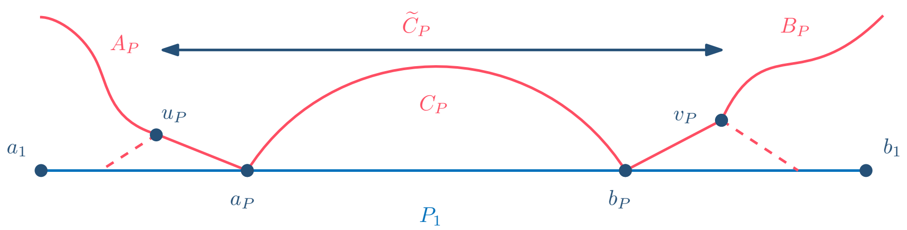


Figure VII.3: The red path represents some path $P \in \mathcal{P}_1$. The blue path is P_1.

We let $\mathcal{A} := \{A_P^{-1} \mid P \in \mathcal{P}_1\} \cup \{B_P \mid P \in \mathcal{P}_1\}$ and $\mathcal{B} := \{\widetilde{C}_P \mid P \in \mathcal{P}_1\}$. In particular, $P_1 \in \mathcal{B}$. Note that by definition, every path in $\mathcal{B}$ is the concatenation of at most 3 paths that are parallel to P_1, and recall that $|\mathcal{P}| \leqslant 2k$, hence we have $|\mathcal{B}| \leqslant 2k$.

We say that the edges of the paths in $\mathcal{A} \cup \mathcal{P}_2$ are *bad*, and we call a vertex $v \in V(G)$ *bad* if it is the endvertex of a bad edge. We call every vertex which is not bad *good*, and we let V_b and V_g denote respectively the set of bad vertices and the set of good vertices of G. Moreover, we set $X_0 := \{a_P \mid P \in \mathcal{P}_1\} \cup \{b_P \mid P \in \mathcal{P}_1\}$ and $X_1 := \{u_P \mid P \in \mathcal{P}_1\} \cup \{v_P \mid P \in \mathcal{P}_1\}$. Note that in particular, $|X_0| \leqslant 4k$ and that the only bad vertices that might belong to P_1 are the vertices a_P, b_P for $P \in \mathcal{P}_1$ whenever they are respectively equal to u_P, v_P, hence $V(P_1) \cap V_b \subseteq X_0$.

We now consider the subgraph $G_0 := \bigcup_{P \in \mathcal{B}} P$ of G. The following remark immediately follows from the definition of $\mathcal{B}$, the fact that $|\mathcal{P}_1| \leqslant |\mathcal{P}| \leqslant 2k$, and that every path in $\mathcal{B}$ is edge-covered by at most 3 paths that are parallel to P_1.

Remark VII.3.6. *G_0 is edge-covered by at most $6k$ paths that are parallel to P_1.*

Lemma VII.3.7. *We have $V_g \subseteq V(G_0)$. Moreover, $G[V_g]$ is a subgraph of G_0.*

Proof. Let u be a good vertex. As G is covered by $\mathcal{P}$, there exists some path $P \in \mathcal{P}$ such that $u \in V(P)$. In particular, as u is a good vertex, we must have $P \in \mathcal{P}_1$, and moreover u must be a vertex of $\widetilde{C}_P$. It implies that $u \in V(G_0)$, and thus that $V_g \subseteq V(G_0)$. The second part of the lemma follows from the fact that by definition, no bad edge can be incident to a good vertex. In particular, note that every edge from $E(G) \setminus E(G_0)$ is bad, hence $G[V_g]$ is indeed a subgraph of G_0. □

Our goal from now on will be to prove the following lemma.

Lemma VII.3.8 (Main)**.** *There exists a set $X \subseteq V(G)$ of size at most $720k^3 + 4k$ that separates every vertex on P_1 from V_b in G.*

Proof of Theorem VII.3.1 using Lemma VII.3.8. Let X be the set given by Lemma VII.3.8. In particular, by Lemma VII.3.7, every component of $G - X$ that contains a vertex of P_1 induces a subgraph of G_0. By Lemma VII.2.2, Lemma VII.3.2 and Remark VII.3.6, G_0 is $6k$-layered, and thus has pathwidth at most $6k$, allowing us to conclude the proof of Theorem VII.3.1. By Remark VII.3.6, the graph G_0 is edge-covered by at most $6k$ paths parallel to P_1, which by Lemma VII.3.4 implies that it has a $6k$-layering, and thus by Lemma VII.2.2, its pathwidth is at most $6k$. □

Reduction to a separation problem in the graph G_0. We will now show that, in order to prove Lemma VII.3.8, it is enough to separate $V(P_1)$ from V_b in the subgraph G_0 of G. We show that Lemma VII.3.8 amounts to prove the following key lemma, that we will prove in Section VII.3.2.

Lemma VII.3.9 (Key lemma)**.** *Let P be a path of G_0 which is parallel to P_1, and $A \in \mathcal{A} \cup \mathcal{P}_2$ which intersects P. Then there exists some set $X_{A,P} \subseteq V(G_0)$ of vertices of size $30k$ such that $X_{A,P}$ separates every vertex of $V(A) \cap V(P)$ from the vertices of P_1 in G_0.*

Proof of Lemma VII.3.8 using Lemma VII.3.9. Recall that every path of $\mathcal{B}$ is the concatenation of at most 3 paths parallel to P_1. We thus consider a set $\mathcal{B}'$ of size at most $3|\mathcal{B}|$ of paths parallel to P_1 that cover the edges of the paths from $\mathcal{B}$. Let I denote the set of all pairs (A, P) such that $A \in \mathcal{A} \cup \mathcal{P}_2$, $P \in \mathcal{B}'$, and $V(A) \cap V(P) \neq \varnothing$.

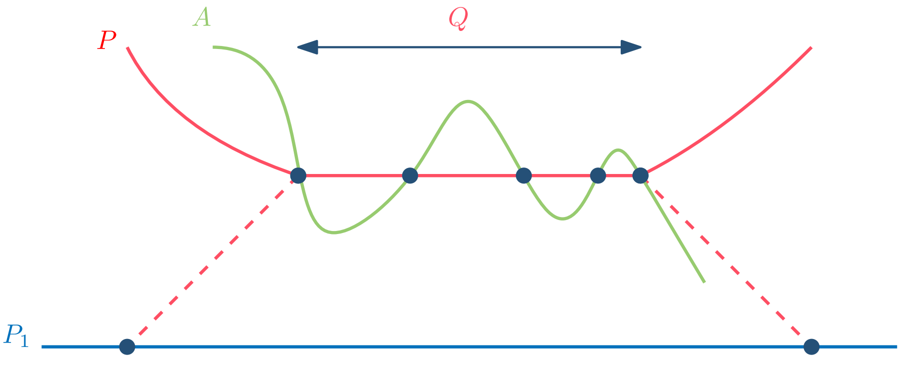


Figure VII.4: Configuration of Lemma VII.3.9.

We set

$$X := X_0 \cup \left(\bigcup_{(A,P)\in I} X_{A,P} \right),$$

where the sets $X_{A,P}$ are given by Lemma VII.3.9. Note that $|\mathcal{A} \cup \mathcal{P}_2| \leqslant 2|\mathcal{P}_1| + |\mathcal{P}_2| \leqslant |\mathcal{P}_1| + |\mathcal{P}| \leqslant 4k$ and $|\mathcal{B}'| \leqslant 3|\mathcal{B}| = 3|\mathcal{P}_1| \leqslant 6k$, thus $|I| \leqslant 24k^2$ and $|X| \leqslant 720k^3 + 4k$.

We now show that X separates $V(P_1)$ from V_b in G. Suppose, for contradiction, that there exists a path R in $G - X$ connecting a vertex v from V_b to a vertex u on P_1. Among those paths, choose a path R with minimum length. Since $V(P_1) \cap V_b \subseteq X_0 \subseteq X$, we have $v \notin V(P_1)$, thus R has length at least 1. Also, since R is a path of minimum length connecting V_b to P_1, every vertex in $R - v$ must be good. Since every vertex in $R - v$ is good, there is no bad edge in R. Hence, as $\mathcal{P}$ edge-covers G, all the edges of R must be covered by the paths from $\mathcal{B}$, implying that R is also a path of G_0.

This implies in particular that $v \in V(G_0) \cap V_b$. Thus, by definition of G_0 and V_b, there exist some paths $P \in \mathcal{B}$ and $A \in \mathcal{A} \cup \mathcal{P}_2$ such that $v \in V(P) \cap V(A)$. Lemma VII.3.9 then implies that R intersects $X_{A,P} \subseteq X$, giving a contradiction. $\square$

VII.3.2 *Proof of Lemma VII.3.9: Don't worry, it's going to be $O(k)$*

We fix some path P in G_0 which is parallel to P_1, and some path $A \in \mathcal{A} \cup \mathcal{P}_2$ which intersects P. Let Q denote the minimal subpath of P containing all vertices in $V(A) \cap V(P)$, and let b, c denote the endvertices of Q such that $b \leqslant_1 c$. We will in fact prove that one can find a set $X_{A,P}$ of vertices with the desired size that separates in G_0 all the vertices of Q from P_1. By Remark VII.3.6, G_0 is coverable by at most $6k$ paths which are all parallel to P_1. In particular, for every $i \geqslant 0$, the set $V_i := \{v \in V(G_0) \mid d_G(v, a_1) = i\}$ contains at most $6k$ vertices, and separates in G_0 the sets $\bigcup_{j \leqslant i} V_j$ and $\bigcup_{j \geqslant i} V_j$. We also may assume without loss of generality that Q contains at least 2 vertices, hence $b <_1 c$ (otherwise, we conclude by choosing $X_{A,P} := V(Q) = \{b\}$).

For every $v \in V(G_0)$, we let $\iota(v)$ denote the unique index $i \geqslant 0$ such that $v \in V_i$. For every path L in G_0 which is parallel to P_1, if u, v denote the endvertices of L with $u \leqslant_1 v$, we define the *projection* of L on P_1 as the set $U_L := \{x \in V(P_1) \mid \iota(u) < \iota(x) < \iota(v)\}$. We say that L is *$\nearrow$-free* if there does not exist in G_0 a path R parallel to P_1 that connects a vertex $x \in U_L$ to a vertex $y \in V(L)$ with $x \leqslant_1 y$ (see Section VII.3.2). Symmetrically, we say that L is *$\searrow$-free* if there does not exist in G_0 a path R parallel to P_1 that connects a vertex $x \in U_L$ to a vertex $y \in V(L)$ with $x \geqslant_1 y$. Our main result in this subsection will be Lemma VII.3.12, which states that if Q is $\nearrow$-free or $\searrow$-free, then we can separate Q from P_1 in G_0 after removing $30k$ vertices. Before proving this, we will first show in the next two lemmas that Q can be covered by at most two subpaths which are each $\nearrow$-free or $\searrow$-free.

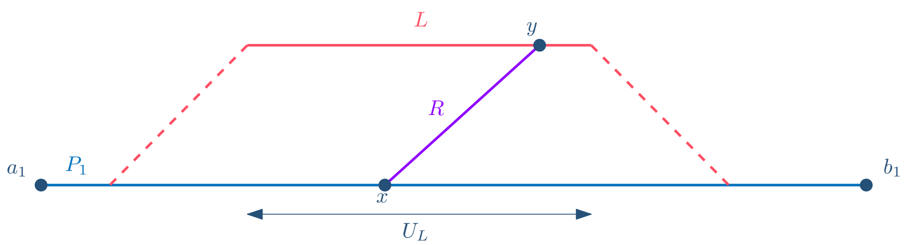


Figure VII.5: The path L (in red) is not $\nearrow$-free.

Lemma VII.3.10. *If $A \in \mathcal{A}$, then Q is $\nearrow$-free or $\searrow$-free.*

Proof. We assume first that $A = A_S^{-1}$ for some $S \in \mathcal{P}_1$, and claim that the case where $A = B_S$ is symmetric. Moreover, we assume that $A[u_S, b]$ is a subpath of $A[u_S, c]$, as depicted in Section VII.3.2, and claim that the case where $A[u_S, c]$ is a subpath of $A[u_S, b]$ is symmetric. We show that in this case, Q must be $\nearrow$-free. As P, and thus Q are parallel to P_1, note that every vertex $v \in V(Q)$ satisfies $\iota(v) \in [\iota(b), \iota(c)]$. Let A' denote the subpath of S^{-1} which starts at a_S and contains A as a subpath.

Assume for sake of contradiction that there exists some path R in G_0 which is parallel to P_1 in G, and which connects a vertex $x \in U_Q$ to a vertex $y \in V(Q)$, with $x \leqslant_1 y$. We consider two cases according to whether $a_S \leqslant_1 x$ or not, described in Section VII.3.2.

If $a_S \leqslant_1 x$, then by Lemma VII.3.4, the path $P_1[a_S, x] \cdot R \cdot Q[y, c]$ is parallel to P_1. On the other hand, recall that by definition of u_S, and as $c \neq u_S$ (this is because $b \neq c$ and $A[u_S, c]$ contains b), $A'[a_S, c]$ cannot be parallel to P_1. In particular, $A'[a_S, c]$ must be strictly longer than $P_1[a_S, x] \cdot R \cdot Q[y, c]$, implying a contradiction as A' is a shortest path in G.

If $a_S >_1 x$, then as $x \in U_Q$, $x >_1 b$. Then, the path $R \cdot Q[y, c]$ is vertical with respect to a_1, and thus strictly shorter than Q. Note in this case, the path $A'[a_S, b]$ is also strictly longer than $P_1[a_S, x]$. It thus implies that $P_1[a_S, x] \cdot R \cdot Q[y, c]$ is strictly shorter than the path $A'[a_S, c]$, contradicting again that A' is a shortest path in G. $\square$

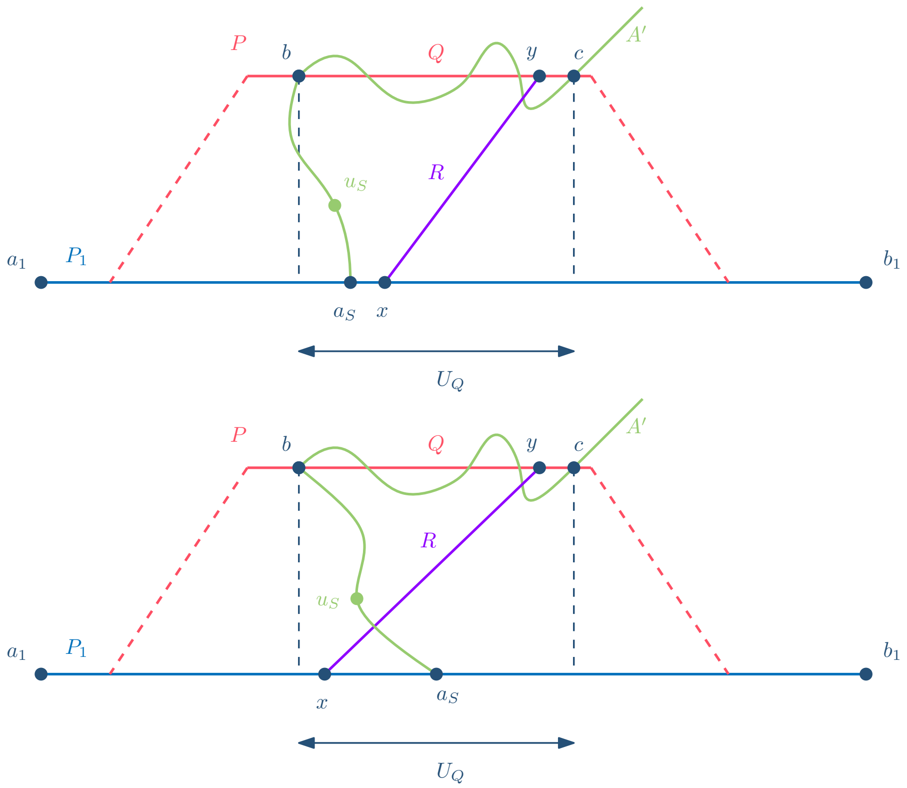


Figure VII.6: Top: configuration in the proof of Lemma VII.3.10 when $a_S \leqslant_1 x$. Bottom: configuration in the proof of Lemma VII.3.10 when $a_S >_1 x$.

Lemma VII.3.11. *If $A \in \mathcal{P}_2$, then there exists subpaths Q_1, Q_2 such that $Q = Q_1 \cdot Q_2$, Q_1 is $\searrow$-free and Q_2 is $\nearrow$-free.*

Proof. If Q is $\nearrow$-free, then we immediately conclude by choosing Q_1 to be the path having b as a single vertex, thus we may assume that there exists some path R parallel to P_1 connecting some vertex $x \in U_Q$ to a vertex $y \in V(Q)$ such that $x \leqslant_1 y$. We moreover choose such a path so that x is maximal with respect to $\leqslant_1$. We let z be the unique vertex of Q such that $\iota(z) = \iota(x)$, and set $Q_1 := Q[b,z]$, $Q_2 := Q[z,c]$. By maximality of x, note that Q_2 must be $\nearrow$-free. We moreover claim that Q_1 is $\searrow$-free. Assume for a contradiction that it is not the case, and that there exists a shortest path R' parallel to P_1 connecting vertices $y' \leqslant_1 x'$, with $y' \in V(Q_1)$ and $x' \in U_{Q_1}$ (see Section VII.3.2). Then the path $T := Q_1[b,y'] \cdot R' \cdot P_1[x',x] \cdot R \cdot Q_2[y,c]$ is a path in G from b to c that intersects P_1 and which is parallel to P_1. In particular, as $b, c \in V(A)$ the existence of T implies that A is a reducing path belonging to $\mathcal{P}$, contradicting our assumption that $\mathcal{P}$ is reduced. □

The next lemma is the crucial part from the proof of Lemma VII.3.9.

Lemma VII.3.12. *Let L be a path in G_0 that is parallel to P_1. If L is $\nearrow$-free or $\searrow$-free, then there exists $X \subseteq V(G_0)$ of size at most $18k$ that separates in G_0 the vertices of L from the ones of P_1.*

Proof. We let b_L, c_L denote the endvertices of L, so that $b_L \leqslant_1 c_L$, and let $X'_0 := V_{\iota(b_L)} \cup V_{\iota(c_L)}$. Recall that by Remark VII.3.6, G_0 is edge-coverable

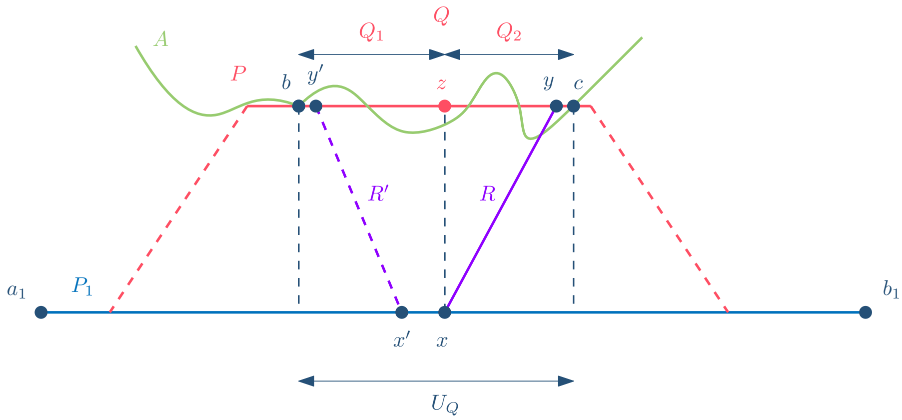


Figure VII.7: The configuration at the end of the proof of Lemma VII.3.11. The purple dashed path R' represents a shortest path which is parallel to P_1, and connects vertices $y' \leqslant_1 x'$, with $y' \in V(Q_1)$ and $x' \in U_{Q_1}$.

by k' paths $Q_1, \dots, Q_{k'}$, all of them parallel to P_1 for some $k' \leqslant 6k$. In particular, it implies that X'_0 has size at most $12k$, and separates L from every vertex x of P_1 such that $x \leqslant_1 b$ or $x \geqslant_1 c$. It remains to separate L from U_L. We assume without loss of generality that L is $\nearrow$-free, the other case being symmetric, up to exchanging the roles of a_1 and b_1. We set $Q_0 := L$, and for each $j \in \{0, \dots, k'\}$, we let $H_j := P_1 \cup Q_0 \cup Q_1 \cup \dots \cup Q_j$. Observe that in particular, $H_0 = P_1 \cup L$ and $H_{k'} = G_0$.

Informally, our proof works as follows: the set X separating P_1 from L that we construct will be the union of X'_0 and of at most one vertex y_j for each path Q_j ($j \in \{1, \dots, k'\}$). We will choose the vertices y_j as follows: we start considering the rightmost (i.e. maximal with respect to $\leqslant_1$) vertex y_1 which is in the intersection of L and of some path Q_{j_1} with $j_1 \in \{1, \dots, k'\}$. Without loss of generality, we may assume that $j_1 = 1$. Note that by maximality of y_1, the set $X'_0 \cup \{y_1\}$ separates the vertices of $L[y_1, c_L]$ from the ones of P_1 in G_0 (and thus also in H_1). Moreover, as L is $\nearrow$-free, none of the vertices from $L \cup Q_1$ which are smaller or equal to y_1 with respect to $\leqslant_1$ can belong to P_1, hence $X'_0 \cup \{y_1\}$ separates the vertices of $L \cup Q_1$ from the ones of P_1 in H_1. In particular, observe that if none of the paths $Q_2, \dots, Q_{k'}$ contains a vertex from $L \cup Q_1$ which is smaller or equal to y_1 with respect to $\leqslant_1$, then it means that $X'_0 \cup \{y_1\}$ separates L from P_1 in G. We may thus assume that there exists some vertex $y_2 \leqslant_1 y_1$ that belongs to $V(Q_{j_2}) \cap V(Q_1 \cup L)$ for some $j_2 \geqslant 2$. Again, we choose such a vertex y_2 maximal with respect to $\leqslant_1$, and assume without loss of generality that $j_2 = 2$. By maximality of y_2, the set $X'_0 \cup \{y_1, y_2\}$ separates the vertices of L which are greater or equal to y_2 from the ones of P_1 in H_2, and as L is $\nearrow$-free, one can check that none of the vertices from $L \cup Q_1 \cup Q_2$ which are smaller than y_2 with respect to $\leqslant_1$ can belong to P_1. We then repeat the construction inductively, by choosing at each step some vertex $y_{j+1} \leqslant_1 y_j$ maximal with respect to $\leqslant_1$ that belongs to $V(L \cup Q_1 \cup \dots \cup Q_j) \cap V(Q_{i_{j+1}})$ for some $i_{j+1} \geqslant j+1$ (without loss of generality, we then assume that $i_{j+1} = j+1$), and claim that at each step, the set $X'_0 \cup \{y_1, \dots, y_{j+1}\}$ separates the vertices of L from the ones of P_1 in H_{j+1}, and that none of the vertices from $L \cup Q_1 \cup \dots \cup Q_{j+1}$ which are smaller than y_{j+1} with respect to $\leqslant_1$ can belong to P_1. See Section VII.3.2 for an example.

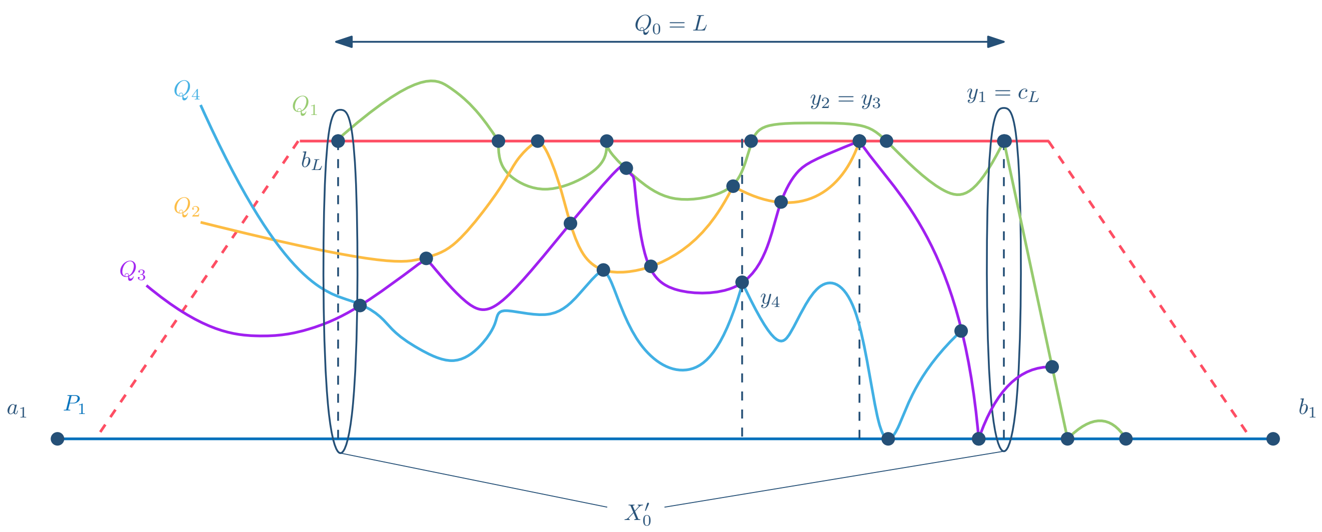


Figure VII.8: In the following example, as Q_2 and Q_3 both have their rightmost intersection with W_1 at the same vertex, we will have $y_2 = y_3$, and we can choose Q_2 and Q_3 in any order in the proof of Lemma VII.3.12.

Formally, we set $y_0 := c_L$ and $W_0 := V(L)$. We construct a sequence of vertices $y_\ell \leqslant_1 \cdots \leqslant_1 y_0$ and a sequence of sets $W_0, \dots, W_\ell$ for some $0 \leqslant \ell \leqslant k'$. These sequences are constructed by induction as follows. Let $j \geqslant 0$ and assume that we already constructed $y_0, \dots, y_j$ and $W_0, \dots, W_j$ for some $j < k'$, such that, up to permuting the indices of the paths $Q_1, \dots, Q_{k'}$, for every $j' \leqslant j$, $y_{j'} \in V(Q_{j'})$. We now let y_{j+1} be any vertex which is maximal with respect to $\leqslant_1$ such that

- $y_{j+1} \leqslant_1 y_j$,
- $y_{j+1} \in W_j \cap V\left(\bigcup_{s \geqslant j+1} Q_s\right)$.

If no such vertex exists, then we stop the construction here and set $\ell := j$. Otherwise, we let y_{j+1} be such a vertex, and, up to permuting the indices of the paths $Q_{j+1}, \dots, Q_{k'}$, we assume that $y_{j+1} \in V(Q_{j+1})$. We write $Q_{j+1} = Q^-_{j+1} \cdot Q^+_{j+1}$, where Q^-_{j+1} (resp. Q^+_{j+1}) denote the subpath of Q_{j+1} containing the vertices which are smaller or equal (resp. greater or equal) to y_{j+1} with respect to $\leqslant_1$. Note that Q^-_{j+1} and Q^+_{j+1} are internally disjoint and share y_{j+1} as an endvertex. We set $W_{j+1} := W_j \cup V(Q^-_{j+1})$. See Section VII.3.2 for an illustration.

We prove by induction over $j \geqslant 0$ the following properties.

(i) There exists a path R_j from y_j to c_L in G_0 which is parallel to P_1;

(ii) for every $y \in W_j$, if $y >_1 y_j$, then $y \notin V\left(\bigcup_{s \geqslant j} Q_s\right)$;

(iii) for every $j' \leqslant j$, $V(Q^+_{j'}) \cap W_j \subseteq \{y_{j'}\}$;

(iv) $W_j \cap U_L = \varnothing$;

(v) $X'_0 \cup \{y_1, \dots, y_j\}$ separates W_j from U_L in H_j.

Note that when $j = 0$, Items (i) to (v) are immediate (note that Item (iv) holds in this case because L is $\nearrow$-free, and thus, P_1 must be internally disjoint from L). We now assume that $0 \leqslant j < k'$, and that Items (i) to (v) hold for any $j' \leqslant j$, and we prove that they are also satisfied for $j+1$.

We start showing Item (i). First, as $y_{j+1} \in W_j$, there exists some $0 \leqslant j' \leqslant j$ such that $y_{j+1} \in V(Q_{j'})$ and $y_{j+1} \leqslant_1 y_{j'}$. We consider the path $R_{j'}$ given by induction hypothesis. In particular, both y_{j+1} and $y_{j'}$ belong to $Q_{j'}$, hence by Lemma VII.3.4, the path $R_{j+1} := Q_{j'}[y_{j+1}, y_{j'}] \cdot R_{j'}$ is parallel to P_1, and connects y_{j+1} to c_L in G_0. This proves Item (i).

We now show Item (ii). Assume for sake of contradiction that there exists $y \in W_{j+1}$ such that $y >_1 y_{j+1}$ and $y \in V\left(\bigcup_{s \geqslant j+1} Q_s\right)$. By definition of W_{j+1}, note that if $y >_1 y_j$, then $y \in W_j$. In particular, by induction hypothesis, if $y >_1 y_j$, then $y \notin V\left(\bigcup_{s \geqslant j} Q_s\right)$, and thus $y \notin V\left(\bigcup_{s \geqslant j+1} Q_s\right)$. We thus have $y_{j+1} <_1 y \leqslant_1 y_j$. However, we now obtain a contradiction with the maximality of y_{j+1} with respect to $\leqslant_1$, as $y \in W_j \cap V\left(\bigcup_{s \geqslant j+1} Q_s\right)$.

Let us now prove Item (iii), that is, that for every $j' \leqslant j+1$, $V(Q^+_{j'}) \cap W_{j+1} \subseteq \{y_{j'}\}$. First, note that if $j' \leqslant j$, as the vertices of $W_{j+1} \setminus W_j$ are exactly the internal vertices of Q^-_{j+1}, and as $y_{j+1} \leqslant_1 y_{j'}$, then the only possible vertex of $Q^+_{j'}$ belonging to $W_{j+1} \setminus W_j$ is $y_{j'}$. In particular, by induction hypothesis we immediately deduce that $V(Q^+_{j'}) \cap W_{j+1} \subseteq \{y_{j'}\}$. It remains to check Item (iii) when $j' = j+1$. Note that it follows from Item (ii), as every vertex of Q^+_{j+1} which is distinct from y_{j+1} is greater than y_{j+1} with respect to $<_1$.

To show Item (iv), assume for sake of contradiction that there exists a vertex x in $U_L \cap W_{j+1}$. By induction hypothesis, $x \notin W_j$, hence x is an internal vertex of Q^-_{j+1}. In particular, $x <_1 y_{j+1}$, thus, as Q_{j+1} is parallel to P_1, the subpath $Q_{j+1}[x, y_{j+1}]$ must be parallel to P_1. We let R_{j+1} be the path from y_{j+1} to c_L given by Item (i). In particular, as $y_{j+1} \leqslant_1 c_L$, Lemma VII.3.4 implies that the path $Q_{j+1}[x, y_{j+1}] \cdot R_{j+1}$ is also parallel to P_1, and connects $x \in U_L$ to $c_L \in V(L)$, with $x \leqslant_1 c_L$, contradicting that L is $\nearrow$-free.

We now prove Item (v), i.e., that $X'_0 \cup \{y_1, \dots, y_{j+1}\}$ separates W_{j+1} from U_L in H_{j+1}. Assume for sake of contradiction that this is not the case. In particular, as by Item (iv), $W_{j+1} \cap U_L = \varnothing$, there should exist an edge yy' in $H_{j+1} - (X'_0 \cup \{y_1, \dots, y_{j+1}\})$, with $y \in W_{j+1}$ and $y' \notin W_{j+1}$. By definition of H_{j+1} and W_{j+1}, as the edge yy' does not belong to W_{j+1}, it should be an edge of some path $Q^+_{j'}$, for some $1 \leqslant j' \leqslant j+1$. In particular, Item (iii) then implies that $y = y_{j'}$, a contradiction as R avoids $\{y_1, \dots, y_{j'}\}$.

We now conclude the proof of Lemma VII.3.12. By definition of ℓ, the sequence $y_0, \dots, y_\ell$ cannot be extended further, that is, for every $\ell < s \leqslant k'$, W_ℓ is disjoint from $\{y \in V(Q_s) \mid y \leqslant_1 y_\ell\}$. We set $X := X'_0 \cup \{y_1, \dots, y_\ell\}$. By Item (v), X separates W_ℓ, and thus L from U_L in H_{j+1}. It remains to show that X also separates W_ℓ from U_L in G_0. Assume for sake of contradiction that it is not the case. In particular, as $Q_1, \dots, Q_{k'}$ edge-cover G_0, there would exist an edge $yy' \in E(Q_s)$

for some $s > \ell$, and some vertices $y \in W_\ell$ and $y' \in V(G_0) \setminus W_\ell$, such that $b_L \leqslant_1 y \leqslant_1 c_L$ and $b_L \leqslant_1 y' \leqslant_1 c_L$. By our above remark, we have $y >_1 y_\ell$. Note that we then immediately obtain a contradiction with Item (ii). This implies that X indeed separates L from P_1 in G_0. Finally, observe that

$$|X| \leqslant |X'_0| + k' \leqslant 3k' \leqslant 18k$$

□

Proof of Lemma VII.3.9. By Lemmas VII.3.10 and VII.3.11, there exist two subpaths Q_1, Q_2 of Q such that $Q = Q_1 \cdot Q_2$, and such that for each $i \in \{1,2\}$, Q_i is either $\nearrow$-free or $\searrow$-free. In particular, by Lemma VII.3.12, there exist two sets X_1, X_2 of size at most $18k$ such that for each $i \in \{1,2\}$, X_i separates in G_0 the set $V(Q_i)$ from $V(P_1)$. In particular, the set $X := X_1 \cup X_2$ then separates $V(Q)$ from $V(P_1)$. Moreover, we claim that when going through the proof of Lemma VII.3.12, one gets that X_1 and X_2 both contain a common layer V_i (for $i := \iota(c_{Q_1}) = \iota(b_{Q_2})$), allowing us to slightly reduce our upper bound on $|X_1 \cup X_2|$ to obtain $|X_1 \cup X_2| \leqslant 30k$ (instead of $36k$). □

VII.4 Exact Bounds for k ⩽ 3

In this section, we prove some optimal upper-bounds on the pathwidth of graphs that are edge-coverable by k shortest paths, when $k \in \{2,3\}$.

> **Theorem VII.1.4.** *Let $k \leqslant 3$, and G be a graph edge-coverable by k shortest paths. Then* $\mathrm{pw}(G) \leqslant k$.

VII.4.1 Two Paths

As a warm-up, we start describing the structure of graphs edge-coverable by 2 shortest paths, implying an immediate proof of Theorem VII.1.4 when $k = 2$.

Skewers A *skewer* is a graph sG for which there exists a sequence of pairwise distinct vertices $x_1, \dots, x_\ell$ ($\ell \geqslant 1$) such that:

- for each $i \in \{1, \dots, \ell-1\}$, the vertex x_i is connected to the vertex x_{i+1} by two internally disjoint paths Q_i, R_i having the same length;
- for each $i \in \{1, \dots, \ell-1\}$, the paths Q_i, R_i are separated from $G - (Q_i \cup R_i)$ by the set $\{x_i, x_{i+1}\}$;
- there exist four internally disjoint paths Q_0, R_0, Q_ℓ, R_ℓ (which do not necessarily have the same length), such that Q_0, R_0 (resp. Q_ℓ, R_ℓ) are separated from $G - (Q_0 \cup R_0)$ (resp. $G - (Q_\ell \cup R_\ell)$) by the set $\{x_1\}$ (resp. $\{x_\ell\}$).

See Section VII.4.1 for an illustration of a skewer. For each $i \in \{0, \ldots, \ell\}$, the graph $C_i := Q_i \cup R_i$ is called the *piece* of x_i. Note that we allow the paths Q_i and R_i to have length 1, in which case the graph C_i is just the edge $x_i x_{i+1}$ (recall that the graphs we consider are simple), while if Q_i and R_i have length at least 2, then C_i is an even cycle. Note that every skewer G has a 2-layering, and thus pathwidth at most 2, and that moreover, for every $i \in \{0, \ldots, \ell\}$, and every pair of vertices a, b with $a \in V(Q_i)$ and $b \in V(R_i)$, there exists a path decomposition of G of width 2 in which there exists a bag equal to $\{a, b\}$.

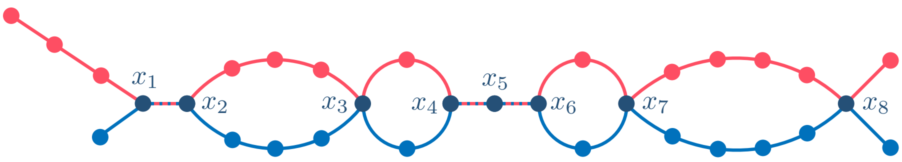


Figure VII.9: A skewer. The blue paths represent the paths R_i and the red paths represent the paths Q_i.

Lemma VII.4.1. *Let P, Q be two shortest paths in a graph G. Then there cannot exist three vertices $a, b, c \in V(P) \cap V(Q)$ such that b is an internal vertex of $P[a, c]$ and such that c is an internal vertex of $Q[a, b]$.*

Proof. Assume for the sake of contradiction that there exist three such vertices $a, b, c \in V(P) \cap V(Q)$. Since b is an internal vertex of $P[a, c]$, $\mathrm{d}(a, b) < \mathrm{d}(a, c)$. But since c is an internal vertex of $Q[a, b]$, $\mathrm{d}(a, c) < \mathrm{d}(a, b)$, a contradiction. □

Proposition VII.4.2. *Every connected graph G which is edge-coverable by two isometric paths is a skewer. In particular,* $\mathrm{pw}(G) \leqslant 2$.

Proof. We let P_1, P_2 be two isometric paths that edge-cover G, and let $x_1, \ldots, x_\ell$ denote the vertices from $V(P_1) \cap V(P_2)$, such that for each $i < j$, x_i is before x_j on P_1. By Lemma VII.4.1, for each $i < j$, x_i is also before x_j on P_2. It thus implies that each x_i separates in G all vertices appearing before it in P_1 and P_2 from all vertices appearing after it in P_1 and P_2. In particular, it easily follows that G is a skewer. □

VII.4.2 Three Paths

This subsection consists in a proof of Theorem VII.1.4 when $k = 3$, which we restate here for convenience.

Theorem VII.4.3. *Let G be a graph which is edge-coverable by 3 isometric paths. Then* $\mathrm{pw}(G) \leqslant 3$.

In the remainder of the subsection, we let G denote a graph which is edge-coverable by three shortest paths P_1, P_2, P_3. We also assume that for each $i \in \{1, 2, 3\}$, P_i is an $a_i b_i$-path for some vertices $a_i, b_i \in V(G)$. Moreover, note that if the P_i's do not all belong to the same connected component of G, then every connected component of G is coverable by at most 2 shortest paths, hence by Proposition VII.4.2, $\mathrm{pw}(G) \leqslant 2$. We thus moreover assume that G is connected.

For two sequences $\mathcal{P}_1 = (X_1, \dots, X_k)$ and $\mathcal{P}_2 = (Y_1, \dots, Y_\ell)$ of vertex subsets such that $X_k = Z = Y_\ell$, we say that $\mathcal{P} = (X_1, \dots, X_k, Z, Y_1, \dots Y_\ell)$ is *the glue of* $\mathcal{P}_1$ and $\mathcal{P}_2$ *by the bag* Z. The following three lemmas deal with some first easy cases.

Lemma VII.4.4. *If the intersection of two of the paths P_1, P_2, P_3 is included in the third one, then* $\mathrm{pw}(G) \leqslant 3$.

Proof. Assume without loss of generality that $V(P_1) \cap V(P_2) \subseteq V(P_3)$, and that G is connected. Note that if one of the two paths P_1, P_2 is disjoint from P_3, then G is not connected, and we get a contradiction, hence both paths intersect P_3. For each $i \in \{1, 2\}$, we let u_i, v_i denote the two vertices from $V(P_i) \cap V(P_3)$ maximizing $d(u_i, v_i)$, and such that $d(a_3, u_i) \leqslant d(a_3, v_i)$, and we let $B_i := P_i[u_i, b_i]$ (see Section VII.4.2). Then the graph $H := P_3 \cup B_1 \cup B_2$ admits a 3-layering $(V_0, \dots, V_r)$, where $r := \mathrm{ecc}_H(a_3)$, and for each $i \in \{0, \dots, r\}$, $V_i := \{u \in V(H) \mid d(a_3, u) = i\}$; this follows from the observation that the three paths P_3, B_1, B_2 are vertical with respect to a_3 (note that the hypothesis that $V(P_3) \subseteq V(P_1) \cap V(P_2)$ is used here). In particular, note that this layering satisfies the property that each path P_j intersects at most once each layer V_i. Observe now that G is obtained from H by attaching the two pendant paths $A_1 := P_1[a_1, u_1]$ and $A_2 := P_2[a_2, u_2]$ to the vertices u_1 and u_2. We claim that it is then not hard to construct a 3-layering of G obtained from the one of H by adding these two pendant paths to the layering $(V_i)_{0 \leqslant i \leqslant r}$. ☐

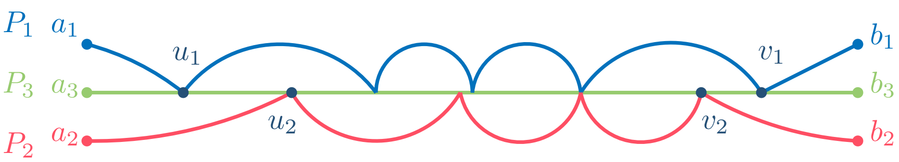


Figure VII.10: Proof of Lemma VII.4.4.

Lemma VII.4.5. *If one of the three paths P_1, P_2, P_3 intersects the other two at most twice in total, then* $\mathrm{pw}(G) \leqslant 3$.

Proof. Assume without loss of generality that we have $|V(P_3) \cap (V(P_1) \cup V(P_2))| \leqslant 2$. Note that if P_3 does not intersect $V(P_1) \setminus V(P_2)$ (or symmetrically, $V(P_2) \setminus V(P_1)$), then we have $V(P_3) \cap V(P_1) \subseteq V(P_2)$ (and symmetrically, $V(P_3) \cap V(P_2) \subseteq V(P_1)$), in which case Lemma VII.4.4 implies that $\mathrm{pw}(G) \leqslant 3$. We may thus assume that P_3 intersects $V(P_1) \cup V(P_2)$ in exactly two vertices $a \in V(P_1) \setminus V(P_2)$ and $b \in V(P_2) \setminus V(P_1)$.

We now set $H := P_1 \cup P_2$. By Proposition VII.4.2, H is a skewer, and we let $x_1, \dots, x_\ell$, $R_0, \dots, R_\ell$, $Q_0, \dots, Q_\ell$, denote its corresponding components with the notations of Section VII.4.1. Assume first that there exists some $i \in \{0, \ell\}$, such that $a \in R_i$ and $b \in Q_i$. Then H has a path decomposition of width 2 that contains a bag equal to $\{a,b\}$. In particular, as $\{a,b\}$ separates $V(P_3)$ from $V(H)$ in G, we can insert after the bag $\{a,b\}$ of this path decomposition some path decomposition of width 1 of P_3 to which we add in each bag the set $\{a,b\}$. This gives a path decomposition of width at most 3 of G.

Assume now that we are not in the previous situation. In particular, as we assumed that $a \in V(P_1) \setminus V(P_2)$ and $b \in V(P_2) \setminus V(P_1)$, up to symmetry, there exist $i \neq j$ such that $a \in R_i$ and $b \in Q_j$. We now let $P_1' := R_0 \cdots R_{j-1} \cdot Q_j \cdot R_{j+1} \cdots R_\ell$ and $P_2' := Q_0 \cdots Q_{j-1} \cdot R_j \cdot Q_{j+1} \cdots Q_\ell$. Observe that H is edge-covered by P_1', P_2', and that P_1', P_2' are shortest paths in G, such that $a, b \in V(P_1')$. In particular, we can again conclude as above, using Lemma VII.4.4 with respect to the paths P_1', P_2', P_3 that edge-cover G. □

Lemma VII.4.6. *If there are at least two vertices that simultaneously belong to the three paths P_1, P_2, P_3, then $\mathrm{pw}(G) \leqslant 3$.*

Proof. Let u and v be two distinct vertices of $V(P_1) \cap V(P_2) \cap V(P_3)$. Without loss of generality, we assume that a_i, u, v, b_i appear in this order on P_i for $i = 1,2,3$. For $i = 1,2,3$, we denote A_i the $a_i u$-subpath of P_i, and B_i the ub_i-subpath of P_i. Then, the graph G admits a 3-layering $(U_\ell, \ldots, U_1, V_0, \ldots, V_r)$ such that $U_i := \{x \in V(A_1) \cup V(A_2) \cup V(A_3) : \mathrm{d}(u,x) = i\}$ and $V_i := \{x \in V(B_1) \cup V(B_2) \cup V(B_3) : \mathrm{d}_H(u,x) = i\}$; this follows from the fact that $\{u\}$ separates the vertices on the A_i's from the vertices on the B_i's in G, and that the A_i's and B_i's are all vertical w.r.t. u. □

We say that a path P *bounces* between two paths Q_1, Q_2 if there exists $i \in \{1,2\}$ such that P has a subpath P' with both endvertices on Q_i, and such that one of the internal vertices of P' intersects Q_{3-i}.

By Lemma VII.4.5, we can assume from now on that P_1 intersects the other paths at least twice, and by Lemma VII.4.6, that at most one of the vertices in these intersections lies simultaneously on the three paths P_1, P_2, P_3. In particular, up to exchanging a_1 and b_1, and P_2 and P_3, we may assume that if u_0 denotes the vertex on P_1 which is the closest to a_1, and belongs to one of the two paths P_2, P_3, then $u_0 \in V(P_1) \cap V(P_2) \setminus V(P_3)$. We set $A_i := P_i[a_i, u_0]$ and $B_i := P_i[u_0, b_i]$ for $i = 1, 2$.

Remark VII.4.7. *Observe that our choice of u_0 implies that $V(A_1) \cap (V(P_2) \cup V(P_3)) = \{u_0\}$.*

By Lemma VII.4.4 and Lemma VII.4.5, we can assume that P_3 intersects $B_1 - u_0$, and at least one of the paths between $A_2 - u_0$ and $B_2 - u_0$, say $B_2 - u_0$ by symmetry. In particular, at least one of the following situations occurs (up to symmetrically exchange the roles of A_2 and B_2)

(1) P_3 bounces between two of the paths A_2, B_2 and B_1,

(2) P_3 does not intersect A_2,

(3) P_3 intersects the three paths A_2, B_2, B_1 in this order,

(4) P_3 intersects the three paths A_2, B_1, B_2 in this order.

To conclude the proof of Theorem VII.1.4, we will prove that $\mathrm{pw}(G) \leqslant 3$ in each of these cases in the following separate lemmas. This is the most technical part of the proof, especially for cases Item (1) and Item (2). The next four lemmas respectively deal with each of the four cases. We start with Item (1).

Lemma VII.4.8. *If P_3 bounces between two of the paths in $\{A_2, B_1, B_2\}$, then $\mathrm{pw}(G) \leqslant 3$.*

Proof. We let $u_0 \in V(P_1) \cap V(P_2) \setminus V(P_3)$ be such that P_3 bounces between two paths in $\{A_2, B_1, B_2\}$. Note that by Lemma VII.4.1, P_3 cannot bounce between A_2 and B_2. We assume that P_3 bounces between B_1 and B_2, and claim that the case where P_3 bounces between B_1 and A_2 is symmetric. Let B_3 be the longest subpath of P_3 with both extremities u_1 and u_3 in $V(B_1) \cup V(B_2)$. Since P_3 bounces between B_1 and B_2, B_3 has length at least 2. We moreover assume without loss of generality that $0 < \mathrm{d}(u_0, u_1) \leqslant \mathrm{d}(u_0, u_3)$ (recall that $u_0 \notin V(P_3)$).

We claim that B_3 is vertical with respect to u_0. This is immediate if B_3 has both endvertices on B_1, or symmetrically on B_2, as both B_1, B_2 are shortest paths in G starting from u_0. Assume now without loss of generality that $u_1 \in V(B_1)$, while $u_3 \in V(B_2)$. Then one can easily see that, as B_3 bounces between B_1 and B_2, there exist two distinct vertices $u_2 \in V(B_2), u_2' \in V(B_1)$ such that u_1, u_2, u_2', u_3 are pairwise distinct and appear in this order on B_3. In particular, in the latter case, $B_3[u_1, u_2']$ and $B_3[u_2, u_3]$ both bounce between B_1 and B_2, and have both endvertices respectively on B_1, and on B_2, so they are both vertical with respect to u_0. As they edge-cover B_3, and share the non-trivial proper subpath $B_3[u_2, u_2']$, it follows that B_3 is also vertical with respect to u_0.

We now prove that B_3 does not intersect A_2. Assume for sake of contradiction that there exists a vertex w in $V(B_3) \cap V(A_2)$. As $u_0 \notin V(P_3)$, $w \neq u_0$. As B_3 bounces between B_1 and B_2, note that $V(B_3) \cap V(B_2)$ contains a vertex v distinct from u_0. By the previous paragraph, observe that the path $Q := B_3[v, w]$ is vertical with respect to u_0, hence $d(v, w) < \max(d(u_0, v), d(u_0, w))$. In particular, we then obtain a contradiction as $P_2[v, w]$ has length $d(u_0, v) + d(u_0, w)$, and should be a shortest vw-path in G.

We now consider the subpaths A_3, C_3 of P_3 such that $P_3 := A_3 \cdot B_3 \cdot C_3$. By construction of B_3, both $A_3 - u_1$ and $C_3 - u_3$ are disjoint from $V(B_1) \cup V(B_2)$.

We prove that C_3 does not intersect A_2. Assume for sake of contradiction that C_3 intersects A_2, on a vertex w (see Section VII.4.2). First, note that B_3 should then intersect B_2 in at most one vertex, as otherwise, the paths P_2, P_3 would contradict Lemma VII.4.1. We then assume that

B_3 intersects B_2 in exactly one vertex u_2. Since B_3 bounces between B_1 and B_2, in particular u_2 is an internal vertex of B_3, and we must have $u_1, u_3 \in V(B_1) \setminus V(B_2)$. We now set $a := d(u_1, u_2), b := d(u_2, u_3), c := d(u_3, w), d := d(u_0, w), e := d(u_0, u_2)$ and $f := d(u_0, u_1)$, so that the situation is as depicted in Section VII.4.2. Since B_3 is vertical with respect to u_0, we have $e = a + f$. Moreover, since both paths $P_2[w, u_2]$ and $P_3[u_2, w]$ are shortest paths, we have $\mathrm{d}(u_2, w) = b + c = e + d$. Furthermore, as $P_3[u_1, w]$ is a shortest path, its length is shorter than the one of $P_2[u_0, w] \cdot P_1[u_0, u_1]$, giving $a + b + c \leqslant f + d$. Combining all previous inequalities then gives $a + e \leqslant f$, and thus $a = 0$, contradicting that u_2 is an internal vertex of B_3. We then obtain, as desired, that C_3 does not intersect A_2, and thus that C_3 only intersects P_1, P_2 in u_3.

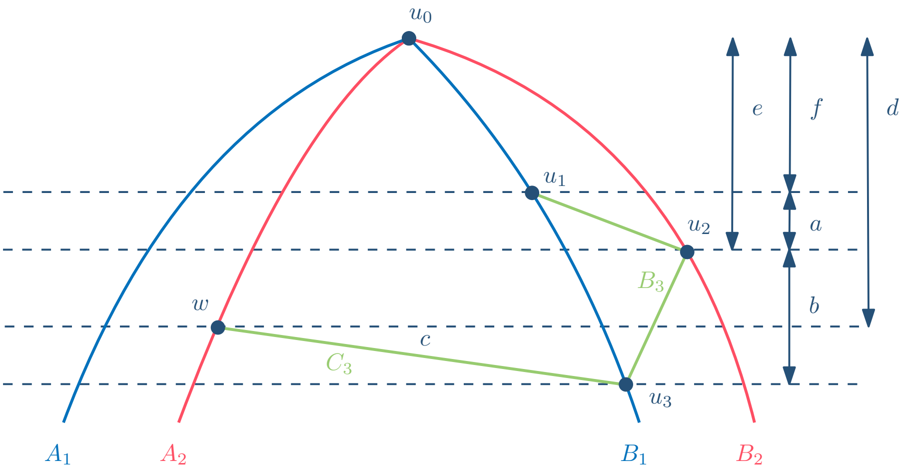


Figure VII.11: Configuration from the proof of Lemma VII.4.8. The dotted lines represent the layers in the BFS from u_0.

We now prove that $V(B_1) \cap V(A_2) = \{u_0\}$. Assume for sake of contradiction that there exists a vertex $v \in V(B_1) \cap V(A_2)$ distinct from u_0. As B_3 bounces between B_1 and B_2 and as $u_0 \notin V(P_3)$, there exist two distinct vertices x, y of B_3 such that $0 < d(x, u_0) < d(y, u_0)$, and such that $x \in V(B_1)$ and $y \in V(B_2)$. We distinguish two cases (depicted in Figure VII.12), according whether $d(u_0, x) \leqslant d(u_0, v)$ or not. Note that in both cases, as A_2, B_1, B_2 and B_3 are vertical with respect to u_0, and as $x, v \neq u_0$, we obtain a contradiction, as the path $B_1[v, x] \cdot B_3[x, y]$ is always strictly shorter than the path $P_2[v, y]$, which is supposed to be a shortest path in G.

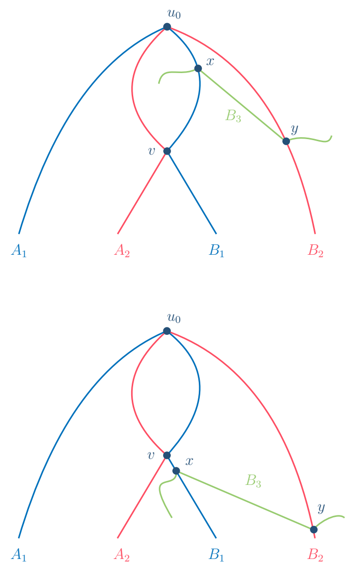


Figure VII.12: The two different cases considered in the proof of Lemma VII.4.8, when we assume that A_2 and B_2 intersect each other. On the left, the situation where $d(u_0, x) \leqslant d(u_0, v)$, and on the right, the situation where $d(u_0, x) \geqslant d(u_0, v)$.

All previous paragraphs together with Remark VII.4.7 thus imply that we can write $G = G_1 \cup G_2$, where $G_1 := A_1 \cup A_2 \cup A_3$ and $G_2 := B_1 \cup B_2 \cup B_3 \cup C_3$, and $\{u_0, u_1\}$ separates G_1 from G_2. We now show that G_1 (resp. G_2) admits a path decomposition of width at most 3 whose rightmost (resp. leftmost) bag contains $\{u_0, u_1\}$. This will immediately imply that $\mathrm{pw}(G) \leqslant 3$.

We first construct such a path decomposition $\mathcal{P}_2$ for G_2. Observe that B_1, B_2 are both vertical with respect to u_0. Moreover, recall that $C_3 - u_3$ is disjoint from $V(B_1) \cup V(B_2)$, hence, as B_3 is vertical with respect to u_0, it implies that $B_3 \cdot C_3$ is also vertical with respect to u_0. It thus implies that the partition $(V_i)_{i \geqslant 0}$ of $V(G_2)$ defined by setting for each

$i \in \{0, \ldots, r\}$ (where $r := \mathrm{ecc}_{G_2}(u_0)$), $V_i := \{x \in V(G_2) \mid d(u_0, x) = i\}$ is a 3-layering of G_2. Moreover, note that if we set $s := d(u_0, u_1)$, then the two subsequences $\mathcal{S}_1 := (V_0, \ldots, V_s)$ and $\mathcal{S}_2 := (V_s, \ldots, V_r)$ of $(V_i)_{0 \leqslant i \leqslant r}$ are respectively 2- and 3-layerings of the subgraphs G_2', G_2'' of G_2 induced respectively by $V_0 \cup \cdots \cup V_s$ and by $V_s \cup \cdots \cup V_r$ (see Section VII.4.2). In particular, two applications of Lemma VII.2.2 give path decompositions $\mathcal{P}_2', \mathcal{P}_2''$ respectively of G_2' and G_2'' such that:

- $\mathcal{P}_2'$ has width at most 2, and its extremities are $V_0 = \{u_0\}$ and V_s,
- $\mathcal{P}_2''$ has width at most 3, and its leftmost bag is V_s (and thus contains u_1).

To obtain a path decomposition $\mathcal{P}_2$ of G_2 with the desired properties, we can now simply add the vertex u_1 in every bag of $\mathcal{P}_2'$, and glue the obtained path decomposition with $\mathcal{P}_2''$.

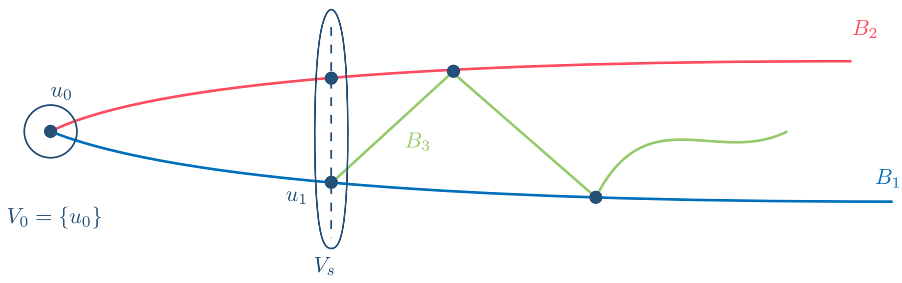


Figure VII.13: The 3-layering of G_2 described at the end of the proof of Lemma VII.4.8.

It remains to construct a path decomposition $\mathcal{P}_1$ of width 3 of G_1 whose rightmost bag contains $\{u_0, u_1\}$. By Remark VII.4.7, $\{u_0\}$ separates the vertices of A_1 from the other vertices of G_1. Moreover, by Proposition VII.4.2, $A_2 \cup A_3$ is a skewer. In particular, it admits a path decomposition $\mathcal{P}_1'$ of width at most 2 whose rightmost bag is $\{u_1\}$. As G_1 is obtained from $A_2 \cup A_3$ after attaching some pendant path A_1 on u_0, the existence of $\mathcal{P}_1$ easily follows. $\square$

The next lemma deals with Item (2).

Lemma VII.4.9. *If P_3 and A_2 are disjoint, then* $\mathrm{pw}(G) \leqslant 3$.

Proof. Suppose that P_3 and A_2 are disjoint. Since by Remark VII.4.7, P_3 does not intersect A_1, by Lemma VII.4.4 we may assume that P_3 intersects both B_1 and B_2. We let v_1 and w_1 (resp. v_2 and w_2) be the vertices of $V(B_1) \cap V(P_3)$ (resp. $V(B_2) \cap V(P_3)$) such that v_1 (resp. v_2) is the closest to u_0 and w_1 (resp. w_2) the farthest. Note that it is possible that $v_1 = w_1$, or $v_2 = w_2$. By Lemma VII.4.8, we may assume that P_3 does not bounce between B_1 and B_2, hence we may assume (up to symmetry) that P_3 intersects B_1, and then B_2. By Lemma VII.4.6, we may moreover assume that there is at most one vertex in $V(P_1) \cap V(P_2) \cap V(P_3)$. We write $P_3 = A_3 \cdot B_3 \cdot C_3$, such that B_3 is the subpath of P_3 from the last vertex of $V(P_1) \cap V(P_3)$ to the first vertex of $V(P_2) \cap V(P_3)$ (note that B_3 might contain only one vertex if $V(P_1) \cap V(P_2) \cap V(P_3)$ is not empty). By Lemma VII.4.1, B_3 has one endpoint in $\{v_1, w_1\}$, and the other in $\{v_2, w_2\}$. Then, let $C_1 := P_1[a_1, v_1]$, $D_1 := P_1[v_1, w_1]$, $E_1 := P_1[w_1, b_1]$ and let $C_2 := P_2[a_2, v_2]$, $D_2 := P_2[v_2, w_2]$, $E_2 := P_2[w_2, b_2]$ (see Section VII.4.2).

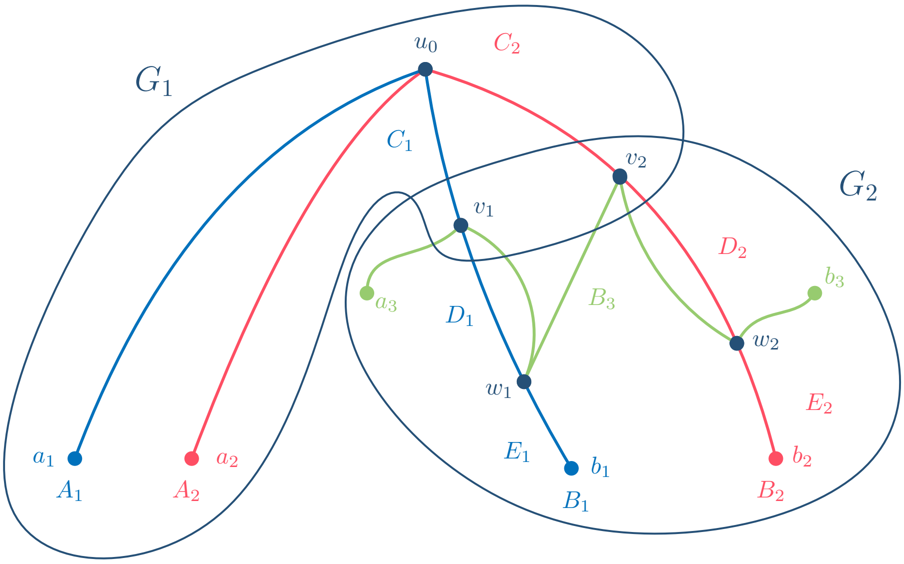


Figure VII.14: Configuration in the proof of Lemma VII.4.9. In this special case, the endvertices of B_3 are v_2, w_1.

In what follows, we let c_1 be the first vertex of P_3 that belongs to $V(P_1) \cup V(P_2)$, and c_2 be the last vertex of P_3 that belongs to $V(P_1) \cup V(P_2)$. As we assumed that P_3 intersects B_1 and then B_2, we have $c_1 \in V(P_1)$ and $c_2 \in V(P_2)$. Moreover, note that if c_1 is in $V(P_2)$, then we have $V(P_3) \cap V(P_1) \subseteq V(P_2)$, and we conclude that $\mathrm{pw}(G) \leqslant 3$ by Lemma VII.4.4. We thus may assume that $c_1 \in V(P_1) \cup V(P_3) \setminus V(P_2)$, and by symmetry that $c_2 \in V(P_2) \cup V(P_3) \setminus V(P_1)$.

Claim VII.4.10. *If one of the two inclusions $V(D_1) \cap V(B_2) \subseteq \{v_1, w_1\}$ or $V(D_2) \cap V(B_1) \subseteq \{v_2, w_2\}$ does not hold, then* $\mathrm{pw}(G) \leqslant 3$.

Proof of the claim. We assume that the first inclusion $V(D_1) \cap V(B_2) \subseteq \{v_1, w_1\}$ does not hold, and claim that the proof if the second inclusion does not hold is symmetric. Then there exists $v \in V(D_1) \cap V(B_2)$ such that $\mathrm{d}(u_0, v_1) < \mathrm{d}(u_0, v) < \mathrm{d}(u_0, w_1)$.

We now claim that we are in the same configuration as in Case (1), with c_2 playing the role of u_0, where P_1 bounces between the paths $P_2[c_2, a_2]$ and $P_3[c_2, a_3]$, as it either intersects v_1, v, w_1 in this order, or w_1, v, v_1 in this order (see Section VII.4.2, left). We thus conclude that $\mathrm{pw}(G) \leqslant 3$ thanks to Lemma VII.4.8. □

Claim VII.4.11. *If $V(E_1) \cap V(C_2) \neq \varnothing$, or $V(E_2) \cap V(C_1) \neq \varnothing$, then* $\mathrm{pw}(G) \leqslant 3$.

Proof of the claim. Assume that there exists a vertex v in $V(E_1) \cap V(C_2)$. We claim that the case where $V(E_2) \cap V(C_1) \neq \varnothing$ is symmetric. The situation is then as depicted in the right of Section VII.4.2.

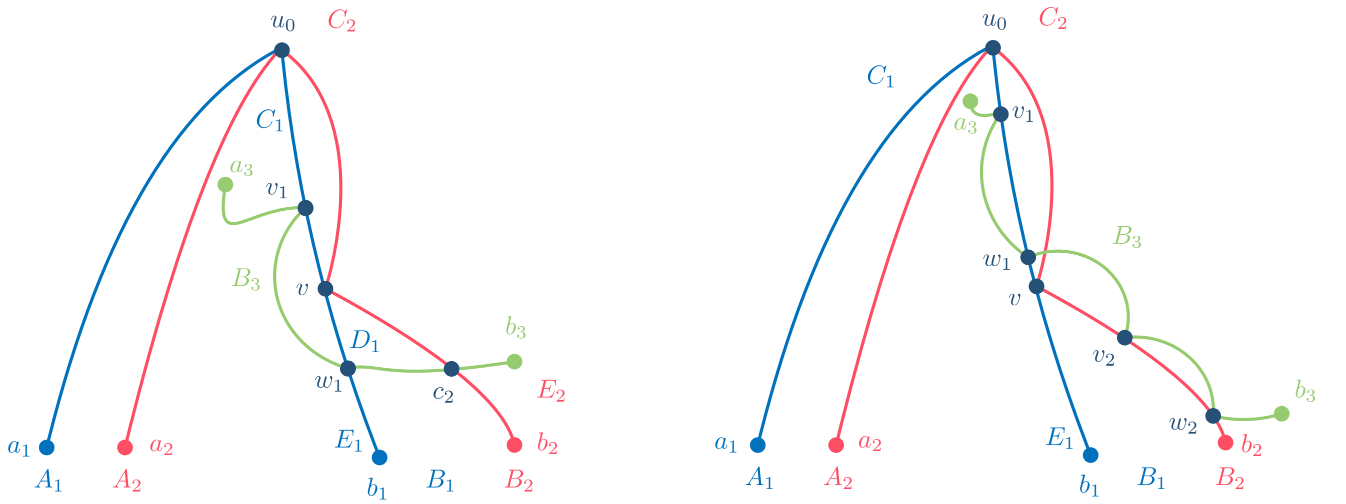


Figure VII.15: Left: configuration of the proof of Claim VII.4.10. In this figure, we depicted the special case where $c_1 = v_1$, however note that there is also a symmetric case where $c_1 = w_1$, in which case c_2 is before v on P_2. Right: configuration of the proof of Claim VII.4.11.

We now set $P_1' := A_1 \cdot P_2[u_0, v] \cdot P_1[v, b_1]$ and $P_2' := A_2 \cdot P_1[u_0, v] \cdot P_2[v, b_2]$. Observe that P_1', P_2' are also shortest paths in G, and that P_1', P_2', P_3 cover the edges of G. In particular, note that $V(P_3) \cap V(P_1') \subseteq V(P_2')$, thus Lemma VII.4.4 applies and gives $\text{pw}(G) \leqslant 3$. □

Now, observe that Claims VII.4.10 and VII.4.11, together with the definition of v_1, v_2 imply that we may assume that the set $\{v_1, v_2\}$ separates in G the subgraph $G_1 := A_1 \cdot C_1 \cup A_2 \cdot C_2$ from the subgraph $G_2 := D_1 \cup E_1 \cup D_2 \cup E_2 \cup P_3$ (see Section VII.4.2). By Proposition VII.4.2, G_1 is a skewer, and admits a path decomposition $\mathcal{P}_1'$ of width at most 2 such that v_1 is in the rightmost bag of $\mathcal{P}_1'$. By adding v_2 to every bag of $\mathcal{P}_1'$, we obtain a path decomposition $\mathcal{P}_1$ of G_1 of width at most 3 such that v_1, v_2 is the rightmost bag of $\mathcal{P}_1$. It thus remains to show that there exists a path decomposition of width 3 of G_2 whose leftmost bag is $\{v_1, v_2\}$.

We now have four distinct cases, two of them being symmetric, depicted on Figure VII.16 according to the order on which the vertices v_1, w_1, v_2, w_2 appear on P_3.

Note that by Claims VII.4.10 and VII.4.11, the only possible common vertices between P_1 and P_2 in G_2 are in $V(D_1) \cap V(D_2)$, and possibly, the vertices v_1, v_2 in the special case when $v_1 = v_2$.

In particular, by definition of w_1, w_2, the set $\{w_1, w_2\}$ separates in G_2 the vertices of the graph $G_3 := D_1 \cup D_2$ from the remaining vertices. By Proposition VII.4.2, and as $u_0 \in V(P_1) \cap V(P_2)$, G_3 admits a path decomposition of width at most 2 whose leftmost bag is $\{w_1, w_2\}$. To conclude our proof, it thus only remains to show in each case that there exists a path decomposition of $G_4 := P_3 \cup E_1 \cup E_2$ of width at most 3, whose leftmost bag is $\{v_1, v_2\}$, and whose rightmost bag is $\{w_1, w_2\}$. We claim that it is then not hard to construct separately in each case such a decomposition. □

The next two lemmas respectively tackle cases (3) and (4).

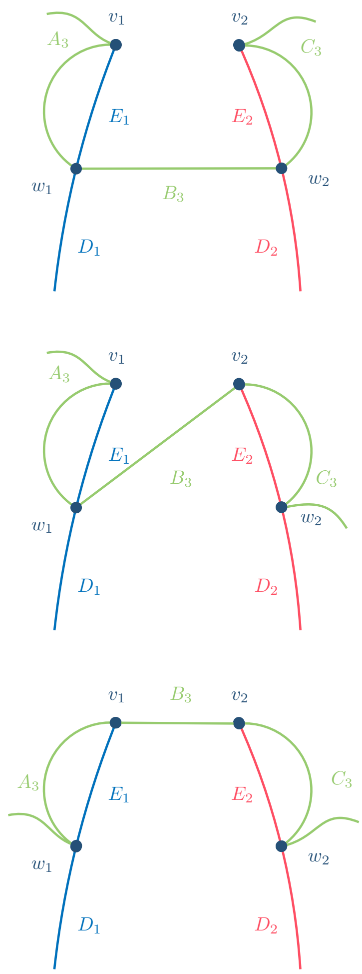


Figure VII.16: Possible ways for the path P_3 (in dark green) to intersect P_1 (in blue) and P_2 (in red) in G_2. Left: the case where v_1, w_1, w_2, v_2 appear in this order on P_3. Middle: the case where v_1, w_1, v_2, w_2 appear in this order on P_3. The case where w_1, v_1, w_2, v_2 appear in this order on P_3 is symmetric. Right: the case where w_1, v_1, v_2, w_2 appear in this order on P_3.

Lemma VII.4.12. *If P_3 intersects the subpaths A_2, B_1, B_2 in this order, then* $\mathrm{pw}(G) \leqslant 3$.

Proof. As $u_0 \notin V(P_3)$, note that P_3 intersects the subpaths $A_2 - u_0, B_1 - u_0, B_2 - u_0$. By Lemma VII.4.1, B_1 cannot intersect both $A_2 - u_0$ and $B_2 - u_0$. By symmetry, assume that B_1 does not intersect $A_2 - u_0$. Let w be the vertex of $V(A_2) \cap V(P_3)$ which is the closest from a_3, and let $A_3 := P_3[a_3, w]$. As B_1 does not intersect $A_2 \setminus w$, and as $u_0 \notin V(P_3)$, we have $w \in V(P_3) \cap V(A_2) \setminus V(B_1)$. Now, observe that if A_3 intersects the path P_1, then as P_3 is disjoint from A_1, it implies that P_3 bounces between A_2 and B_1, in which case Lemma VII.4.8 applies and gives $\mathrm{pw}(G) \leqslant 3$. Assume now that A_3 and P_1 are disjoint. Then, we claim that we are in the configuration of Case (2), with w playing the role of u_0, P_1 playing the role of P_3, and P_3 playing the role of P_1. In particular, Lemma VII.4.9 then applies and gives $\mathrm{pw}(G) \leqslant 3$. □

Lemma VII.4.13. *If P_3 intersects the subpaths A_2, B_2, B_1 in this order, then* $\mathrm{pw}(G) \leqslant 3$.

Proof. As $u_0 \notin V(P_3)$, note that P_3 also intersects the subpaths $A_2 - u_0, B_2 - u_0, B_1 - u_0$. We let x, y, z be three distinct vertices of P_3 such that $x \in V(A_2), y \in V(B_2)$ and $z \in V(A_1)$, and such that y is an internal vertex of $P_3[x, z]$.

Note that by Lemma VII.4.1, no vertex of $P_3[y, b_3]$ belongs to A_2, and no vertex of $P_3[a_3, x]$ belongs to B_2. Moreover, by Lemma VII.4.8, no vertex on $P_3[z, b_3]$ belongs to $P_2 - z$. In particular, it implies that we may assume that x is the vertex from $V(A_2) \cap V(P_3)$ that minimizes the distance $d(a_2, x)$, and that y is the vertex from $V(B_2) \cap V(P_3)$ that minimizes the distance $d(b_2, y)$.

Assume first that B_1 does not intersect $B_2 - u_0$. Note that by Remark VII.4.7, it implies that y is in $V(P_2) \cap V(P_3) \setminus V(P_1)$. We set $D_2 := P_2[y, b_2]$ and $D_3 := P_3[a_3, y]$. By our minimality assumption, y is must then be the only intersection of D_2 with $P_1 \cup P_3$. By Lemma VII.4.8, no vertex on D_3 can belong to B_1. Hence, we have that P_1 does not intersect D_2 nor D_3. By Lemma VII.4.9, with y playing the role of u_0, P_1 playing the role of P_3 and D_2 playing the role of A_1, we obtain that $\mathrm{pw}(G) \leqslant 3$.

We now assume that B_1 intersects $B_2 - u_0$. In particular, B_1 does not intersect $A_2 - u_0$ by Lemma VII.4.1. By our choice of x and Remark VII.4.7, observe that $C_2 := P_2[a_2, x]$ only intersects $P_1 \cup P_3$ in x. We now set $A_3 := P_3[a_3, x]$. Note that by Lemma VII.4.1 and our choice of x, A_3 only intersects P_2 in x. Moreover, note that as A_3 is disjoint from A_1, if A_3 intersects B_1, then P_3 bounces between B_1 and A_2, so by Lemma VII.4.8 we have $\mathrm{pw}(G) \leqslant 3$. We may thus assume that x is the vertex in the intersection of A_3 and $P_1 \cup P_2$. Note that we are then in the situation of Lemma VII.4.9, with x playing the role of u_0, P_1 playing the role of P_3 and C_2, A_3 playing the respective roles of A_1, A_2. We thus conclude that $\mathrm{pw}(G) \leqslant 3$. □

Proof of Theorem VII.1.4. As explained above, observe that, up to replacing P_2 by P_2^{-1}, one of the four cases between (1), (2), (3) and (4) should occur. We thus conclude that $\mathrm{pw}(G) \leqslant 3$ after applying according accordingly Lemma VII.4.8, Lemma VII.4.9, Lemma VII.4.12 and Lemma VII.4.13. □

VII.5 Coverings by Isometric Subtrees

Treewidth. A *tree decomposition* of a graph G is a pair $(T, \mathcal{V})$ where T is a tree and $\mathcal{V} = (V_t)_{t \in V(T)}$ is a family of subsets V_t of $V(G)$ such that:

- $V(G) = \bigcup_{t \in V(T)} V_t$;
- for all nodes t, t', t'' such that t' is on the unique path of T from t to t'', $V_t \cap V_{t''} \subseteq V_{t'}$;
- every edge $e \in E(G)$ is contained in an induced subgraph $G[V_t]$ for some $t \in V(T)$.

The subsets V_t are called the *bags* of the tree decomposition $(T, \mathcal{V})$, and the tree T is called the *decomposition tree*. The *width* of a tree decomposition is the maximum size of a bag minus 1. The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$ is defined as the minimum possible width of a tree decomposition of G. Note that path decompositions of G correspond exactly to the tree decompositions of G whose decomposition tree is a path. In particular, we have in general $\mathrm{tw}(G) \leqslant \mathrm{pw}(G)$.

Recall that a graph G is 2-*connected* if it is connected of order at least 3, and has no *cutvertex*, that is, no vertex v such that $G - v$ is not connected anymore. The *blocks* of G are the inclusion-wise maximal sets of vertices $X \subseteq V(G)$ such that $G[X]$ is connected and has no cutvertex[5]. The following is a well-known and easy fact.

[5] Note that here, blocks are not necessarily 2-connected, as we allow them to have order less than 3.

Proposition VII.5.1 (folklore). *The treewidth of a graph is equal to the maximum treewidth of its blocks (maximal 2-connected components). Put differently, for every $v \in V(G)$, if $X_1, \ldots, X_k$ denote the connected components of $G - v$, then* $\mathrm{tw}(G) = \max_{1 \leqslant i \leqslant k} \mathrm{tw}(G[X_i \cup \{v\}])$.

The following remark is immediate.

Remark VII.5.2. *Let G be a graph edge-coverable by k isometric trees, and X be a block of G. Then $G[X]$ is also edge-coverable by k isometric trees.*

We now prove our main result in this section, which we restate.

Theorem VII.1.5. *Let G be a graph which is edge-coverable by 2 isometric trees. Then* $\mathrm{tw}(G) \leqslant 2$.

Proof. Let T_1, T_2 denote two isometric subtrees of G that cover all its edges. By Proposition VII.5.1 and Remark VII.5.2, we may assume without loss of generality that G is 2-connected. In particular, G has no vertex of degree 1. Moreover, up to removing some edges incident to the leaves of T_1 and T_2, we may further assume that every edge of T_1 which is incident to a leaf of T_1 is not an edge of T_2. Let u be a leaf of T_1. As G has no vertex of degree 1, u must also be a vertex of T_2.

Claim VII.5.3. *Every edge of G is vertical with respect to u.*

Proof of the Claim. Suppose for a contradiction that there exists some edge $vw \in E(G)$ which is not vertical with respect to u. Then it implies that v and w belong to the same layer of the BFS-layering starting from u, that is, $d_G(u,v) = d_G(u,w)$. We let $i \in \{1,2\}$ be such that $vw \in E(T_i)$. As T_i is an isometric subtree of G, we then also have $d_{T_i}(u,v) = d_{T_i}(u,w)$. We then obtain a contradiction as such an edge vw would create a cycle in T_i. □

We now consider the subtree T_1' of T_1 formed by taking the union of all paths P between u and a vertex of T_1 such that all internal vertices of P are in $V(T_1) \setminus V(T_2)$. As G is 2-connected, all leaves of T_1' are in T_2. Let G_v be the subgraph of G induced by the descendants of v, i.e., by vertices x for which there exists some ux-path P which contains v and which is vertical with respect to u. Note that in particular, $v \in V(G_v)$.

Claim VII.5.4. *Each leaf v of T_1' which is distinct from u separates G_v from the rest of G.*

Proof of the Claim. Let v be a leaf of T_1' distinct from u, and suppose for a contradiction, that v does not separate G_v from $G - G_v$. Then there exists some edge $e = xy \in E(G)$ with $x \in V(G_v) \setminus \{v\}$ and $y \in V(G) \setminus V(G_v)$. By Claim VII.5.3, e is vertical with respect to u. As we assumed that $y \notin V(G_v)$, we must have $d_G(u,y) < d_G(u,x)$. Let $i \in \{1,2\}$ be such that $xy \in E(T_i)$. We let P denote the path in T_i connecting u to x, which, by Claim VII.5.3, is vertical with respect to u and thus contains y. As y is not in G_v, observe that P cannot contain v. See Figure VII.17 for the case where $i = 2$. We let w be the least common ancestor in T_i of v and x (recall that by definition of T_1', v is in $V(T_1) \cap V(T_2)$). As P avoids v, we have $v \neq w$. As T_i is an isometric subgraph of G, we then have $d_G(v,x) = d_{T_i}(v,x) = d_{T_i}(v,w) + d_{T_i}(w,x) > d_{T_i}(w,x) = d_G(w,x)$. Now, recall that $x \in V(G_v)$, hence there exists some ux-path in G from v to x going through v, which is vertical with respect to u. In particular, as w is an ancestor of v in T_i, it then implies that $d_G(v,x) \leqslant d_G(w,x)$, giving a contradiction (see Figure VII.17). □

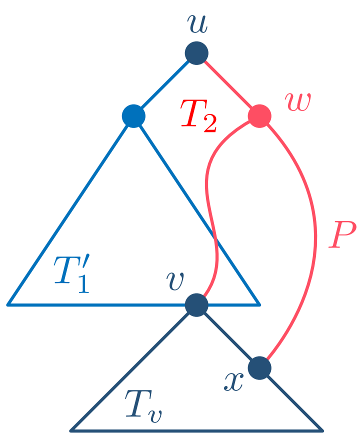


Figure VII.17: Proof of Claim VII.5.4 in the case where $xy \in E(T_2)$.

As G is 2-connected, Claim VII.5.4 implies that for each leaf v of T_1' distinct from u, we have $V(G_v) = \{v\}$. In particular, as we initially chose u to be a leaf of T_1, it then implies that $T_1 = T_1'$, so the only vertices of T_1 that also belong to T_2 are its leaves. Observe that up to reproducing symmetrically the same reasoning, where the roles of T_1 and T_2 are exchanged, we may moreover assume that the only vertices of T_2 belonging to T_1 are its leaves. In other words, we are left with the case where T_1 and T_2 have no common internal nodes, and intersect exactly at their leaves.

In the remainder of the proof, we will say that a graph H has a *mirror-decomposition* if there exist trees T, T' such that

- $H = T \cup T'$;
- the trees T and T' have no common internal nodes;
- there exists a graph isomorphism ι from T to T' such that for every $v \in V(T) \cap V(T')$, we have $\iota(v) = v$.

The proof of Theorem VII.1.5 will be an immediate consequence of the following two Claims.

Claim VII.5.5. *G has a mirror-decomposition, with respect to the trees T_1 and T_2.*

Proof of the Claim. First, observe that as T_1, T_2 are both isometric subgraphs of G, for every two vertices a, b in $V(T_1) \cap V(T_2)$, it holds that $d_G(a,b) = d_{T_1}(a,b) = d_{T_2}(a,b)$. In particular, the claim is then immediate if T_1 and T_2 have at most 2 leaves. We now assume that T_1, T_2 have more than 2 leaves.

We recall that a basic property of trees is that for every three nodes a, b, c in a tree T, there exists a unique *median*, that is, a unique vertex $m(a,b,c)$ of T that belongs to shortest paths between each pair of a, b and c. Note that in particular, if a, b, c are pairwise distinct leaves of a tree T, then their median is an internal node of T.

In what follows, for every $i \in \{1,2\}$ and every three leaves of T_i (and thus also of T_{3-i}), we let $m_i(a,b,c) \in V(T_i)$ denote their median in the tree T_i. We first claim that for every $x \in \{a,b,c\}$, we have $d(m_1(a,b,c),x) = d(m_2(a,b,c),x)$. To see this, fix a, b, c and for each $x \in \{a,b,c\}$, set $d_i(x) := d_{T_i}(m_i(a,b,c),x)$. Note first that as T_1, T_2 are isometric subgraphs in G, we have for every two distinct vertices $x, y \in \{a,b,c\}$ that $d_1(x) + d_1(y) = d_{T_1}(x,y) = d_{T_2}(x,y) = d_2(x) + d_2(y)$. In particular, we then have

$$\begin{aligned} 2(d_1(a) + d_1(b) + d_1(c)) &= d_{T_1}(a,b) + d_{T_1}(b,c) + d_{T_1}(a,c) \\ &= d_{T_2}(a,b) + d_{T_2}(b,c) + d_{T_2}(a,c) \\ &= 2(d_2(a) + d_2(b) + d_2(c)). \end{aligned}$$

By a previous observation, $d_1(b) + d_1(c) = d_2(b) + d_2(c)$, hence in particular it implies that $d_1(a) = d_2(a)$. The equalities $d_1(b) = d_2(b)$ and $d_1(c) = d_2(c)$ are obtained identically.

To conclude the proof of Claim VII.5.5, we now show that for every tree T, the knowledge for each triple (a,b,c) of distinct leaves of the tuples

$$(d(m(a,b,c),a), d(m(a,b,c),b), d(m(a,b,c),c))$$

can be used to reconstruct T uniquely. In other words, for a fixed set L of size at least 3 and a fixed mapping $\varphi : L^3 \to \mathbb{N}^3$, there exists at most one tree T (up to relabelling the internal nodes) such that the leaves of T are exactly the elements of L, and such that for every triple (a, b, c) of distinct leaves, we have

$$\varphi(a, b, c) = (d(m(a, b, c), a), d(m(a, b, c), b), d(m(a, b, c), c)).$$

When such an equality holds for all triple of distinct leaves, we say that T *realizes* φ. Note that by previous paragraphs proving such a statement will immediately conclude our proof of Claim VII.5.5. We show this statement by induction on the size of L. If $|L| = 3$, then note that the only possible tree realizing some given triple $\varphi(a, b, c) = (d_a, d_b, d_c)$ is a subdivided star with three branches of size d_a, d_b and d_c respectively connecting a, b and c to the center. Assume now that $|L| \geqslant 4$ and that the induction hypothesis holds. Suppose that we are given some tree T with set of leaves L realizing some fixed mapping $\varphi : L^3 \to \mathbb{N}^3$. Let a be a leaf of T, and let P be the shortest possible path in T connecting a to a vertex x of degree at least 3 (such a vertex always exists as soon as T has at least 3 leaves). Then $T - P$ is a tree whose set of leaves is exactly $L \setminus \{a\}$. In particular, applying induction hypothesis on $L \setminus \{a\}$ implies that $T - P$ is the unique possible tree realizing the restriction of φ on $(L \setminus \{a\})^3$. Moreover, as x has degree at least 3 in T, there exist two distinct leaves b, c of $T - P$ such that $x = m(a, b, c)$ in T. Write $\varphi(a, b, c) = (d_a, d_b, d_c)$ and observe now that x is necessarily the only vertex of $T - P$ that lies on the bc-path of $T - P$ such that $d(x, b) = d_b$ and $d(x, c) = d_c$. In particular, it implies that the only possibility for a tree to realize φ is to be obtained from $T - P$ after attaching some path P' of length d_a connecting a to x. This concludes the proof. □

Claim VII.5.6. *Every graph admitting a mirror-decomposition has treewidth at most* 2.

Proof of the Claim. We let H be a graph with a mirror-decomposition and T, T', ι be as in the definition of mirror-decomposition. We prove by induction on $|V(T)| = |V(T')| \geqslant 1$ that there exists a tree decomposition of H of width at most 2, and such for each $v \in V(T)$, there exists a bag that contains $\{v, \iota(v)\}$. If $|V(T)| \leqslant 2$, then $|V(H)| \leqslant 3$ (unless $V(T)$ and $V(T')$ are disjoint, in which case we also conclude immediately), and one can conclude by considering the tree decomposition with a single bag equal to $V(H)$. Assume now that $|V(T)| = |V(T')| \geqslant 3$. In particular, there exists some node $u \in V(T)$ such that $T - u$ is not connected. Then, $\{u, \iota(u)\}$ forms a separator of size 2 in H (see Section VII.5).

Now, observe that for each component C of $T - u$, the graph $H_C := H[C \cup \iota(C) \cup \{u, \iota(u)\}]$ admits a mirror-decomposition with respect to the trees $T[C \cup \{u\}]$ and $T'[\iota(C \cup \{u\})]$. By choice of u, the tree $T[C \cup \{u\}]$ has strictly fewer vertices than T, thus the induction hypothesis implies that H_C admits a tree decomposition $(T_C, \mathcal{V}_C)$ such that for every $x \in C \cup \{u\}$, there is a bag containing $\{x, \iota(x)\}$. We let $t_C \in V(T_C)$ be a node of T_C whose bag contains $\{u, \iota(u)\}$. We now conclude the proof of the claim by considering the tree decomposition $(\widetilde{T}, \widetilde{\mathcal{V}})$ of H, where $\widetilde{T}$ is obtained after taking the disjoint union of the trees T_C, and adding a vertex z connected to all nodes t_C, with associated bag $\{u, \iota(u)\}$. □

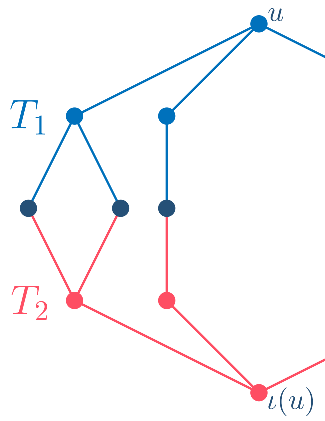


Figure VII.18: Example of mirrored trees. Notice that $\{u, \iota(u)\}$ indeed forms a separator of the graph.

□

VII.6 A linear lower bound

In this section, we prove a general lower bound on the pathwidth of graphs edge-coverable by k shortest paths.

Proposition VII.1.3. *For any integer $k \geqslant 1$, there is a graph G_k that is*

(a) *edge-coverable by k shortest paths, and*

(b) *that has pathwidth at least k.*

The remainder of the section is dedicated to the proof of Proposition VII.1.3. We will construct inductively a sequence $(G_k)_{k \geqslant 1}$ of graphs satisfying the outcome of Proposition VII.1.3.

Construction of the sequence $(G_k)_{k \geqslant 1}$. We first consider the sequence $(\ell_k)_{k \geqslant 1}$ of integers defined inductively by setting $\ell_1 := 3, \ell_2 := 7$ and $\ell_{k+1} := 2(\ell_k + 3)$ for each $k \geqslant 2$. We now construct the sequence $(G_k)_{k \geqslant 1}$ inductively, maintaining at each step the property that G_k admits a k-layering with ℓ_k layers, in which both the left and right extremities contain a single vertex, respectively denoted u_k and v_k.

First, we define separately G_1 and G_2: G_1 is the path of length 3, with extremities u_1, v_1, and G_2 is the cycle of size 14, seen as the union of two internally disjoint $u_2 v_2$-paths of length 7 (see Section VII.6).

We now let $k \geqslant 2$ and construct inductively G_{k+1}. First, we let H, H' denote two disjoint copies of G_k, such that a_k, b_k (resp. a'_k, b'_k) denote the copies of the vertices u_k, v_k in H (resp. in H'). To obtain G_{k+1}, we moreover add an edge between b_k and a'_k, two vertices u_{k+1} and v_{k+1}, two paths A_k, B_k of length 2 respectively between u_{k+1} and a_k, and between

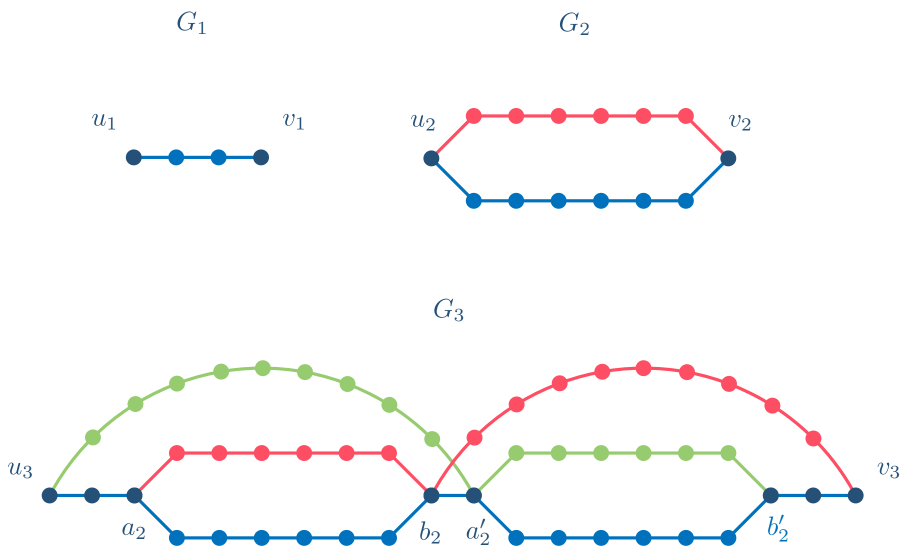


Figure VII.19: The graphs G_1, G_2 and G_3. The colored paths form a covering by k shortest paths for each $k \in [3]$.

v_{k+1} and b'_k, and finally, two paths C_k, D_k of length $l_k + 3$ respectively between u_{k+1} and a'_k, and between v_{k+1} and b_k, whose internal vertices are distinct from all the ones that were present so far (see Section VII.6).

An easy induction on k shows that G_k admits a k-layering L_k containing ℓ_k layers for each $k \geqslant 1$, whose extremities are $\{u_k\}$ and $\{v_k\}$.

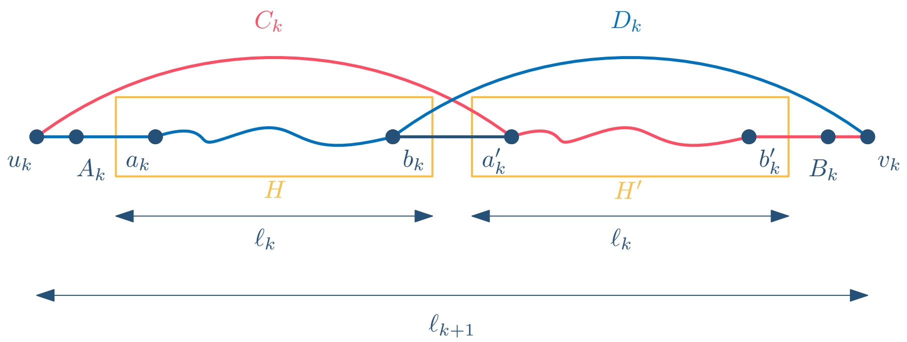


Figure VII.20: Recursive construction of G_{k+1}. The paths Q_k, Q_{k+1} are respectively represented in red and blue.

Proof of Item (a) in Proposition VII.1.3. We show by induction that G_k is edge-coverable by k shortest paths. We will moreover show that one can choose all of these paths to be $u_k v_k$-paths. In particular, it implies that they all contain exactly one vertex in each layer of L_k. The result is immediate for $k \leqslant 2$, so we now let $k \geqslant 2$ and assume that it holds for G_k. We let $P_1, \ldots, P_k$ and $P'_1, \ldots, P'_k$ denote respectively paths that edge-cover H and H', such that for each $i \in [k]$, P_i (resp. P'_i) is a shortest $a_k b_k$-path in H (resp. a shortest $a'_k b'_k$-path in H'). For each $i \in [k-1]$, we set $Q_i := A_i \cdot P_i \cdot b_k a_k \cdot P'_i \cdot B_i$. Moreover, we set $Q_k := C_k \cdot P'_k \cdot B_k$ and $Q_{k+1} := A_k \cdot P_k \cdot D_k$ (see Section VII.6). It is easy to check that $Q_1, \ldots, Q_k$ edge-cover G_{k+1}. Moreover, note that each Q_i has at most one vertex in each layer of L_{k+1}, implying in particular that it is a shortest $u_{k+1} v_{k+1}$-path, which concludes the proof.

Proof of Item (b) in Proposition VII.1.3. We now show that for each $k \geqslant 1$, we have $\mathrm{pw}(G_k) \geqslant k$, which will prove Item (b) in Proposition VII.1.3 and therefore complete its proof.

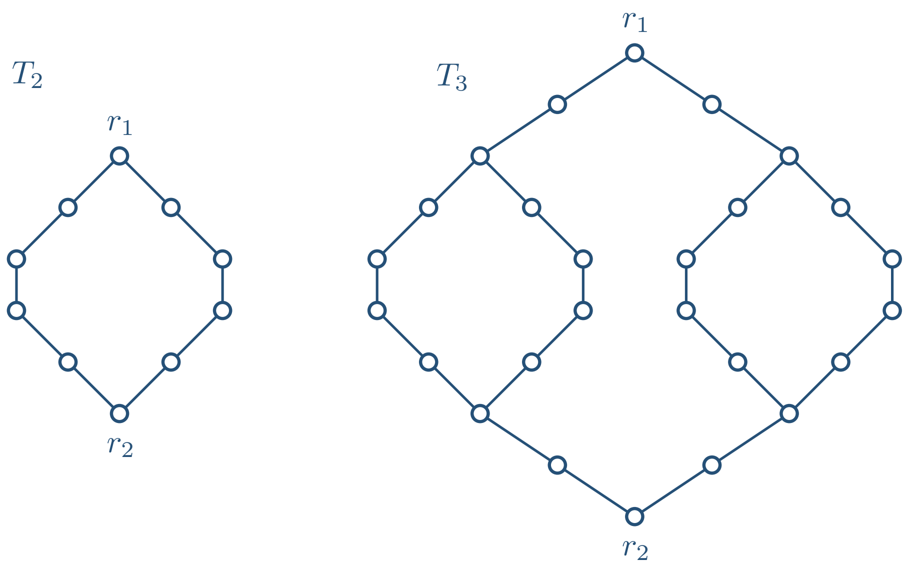


Figure VII.21: The graphs T_2 and T_3.

For each $k \geqslant 1$, we let T_k denote the graph obtained as follows: take two disjoint copies B_1 and B_2 of a complete binary tree of height $k-1$, with respective roots r_1, r_2, subdivide all edges once, and add a matching connecting each leaf of B_1 to its copy in B_2 (see Section VII.6). Note first that for each $k \geqslant 1$, T_k is a minor of G_k. To see this, observe that an easy induction on k further gives that there exists for each $k \geqslant 1$ a model of T_k in which the branch sets of the respective roots of the copies B_1, B_2 of the binary tree are exactly $\{u_k\}$ and $\{v_k\}$ (note also that the edge $b_k a'_k$ can be ignored for this part, i.e. T_k is also a minor of $G_k - a_k b_k$). In particular, as T_k is a minor of G_k, we have for each $k \geqslant 1$ that $\mathrm{pw}(G_k) \geqslant \mathrm{pw}(T_k)$. It thus only remains to show that T_k has pathwidth at least k.

Lemma VII.6.1. *For any integer k, the graph T_k has pathwidth at least k.*

To prove Lemma VII.6.1, we use the game characterization of pathwidth, in terms of the *node searching number* (see Section VI.2 for definitions). A classic result of Kirousis and Papadimitriou [KP85] states that for every graph G, we have $\mathrm{ns}(G) = \mathrm{pw}(G) + 1$. In particular, in order to conclude our proof of Proposition VII.1.3, it will be enough to show that for each $k \geqslant 1$, $\mathrm{ns}(T_k) \geqslant k + 1$.

In order to show this, we need to slightly extend the definition of the game by allowing some winning positions for the robber. For a set $W \subseteq V(G)$ of vertices, which we call a *winning state*, a search $C_1, \ldots, C_t$ on G is *W-winning* if there exists some $i \in [t]$ for which $F_i \cap W \neq \varnothing$. The *$W$-node searching number* of G, denoted $\mathrm{ns}_W(G)$ is the minimum possible integer $k \geqslant 0$ for which G admits a search which is neither W-winning, nor non-evasive. More informally, $\mathrm{ns}_W(G) > k$ means that when playing with k cops, whatever strategy the cops choose, the robber will be able to find accordingly a strategy allowing him either to enter W at some step, or to escape them forever. Combined with all previous observations, next lemma will immediately conclude our proof of Proposition VII.1.3. In what follows, for each $k \geqslant 1$, we let T'_k denote the graph obtained from T_k after adding two new vertices s_1, s_2 of degree 1 respectively attached to the roots r_1, r_2 of the binary trees used in the definition of T_k.

Lemma VII.6.2. *We have:*

1. $\mathrm{ns}(T_k) \geqslant k+1$ *for every* $k \geqslant 1$, *and*
2. $\mathrm{ns}_{\{s_1,s_2\}}(T'_k) \geqslant k+2$ *for every* $k \geqslant 2$.

Proof. The proof goes by induction on $k \geqslant 1$. The base case $k = 1$ is immediate, as T_1 is then an edge with extremities r_1, r_2, in which case it is easy to see that one cop can never catch the robber [6], implying Item 1. T_2 is a cycle of size 10 on which r_1 and r_2 are two antipodal vertices. In this case, it is not hard to check that a robber will always be able to escape to two cops in the node searching game showing Item 1 when $k = 2$. We now show that Item 2 also holds when $k = 2$. For this, we describe a winning strategy for the robber in the $\{s_1, s_2\}$-node searching game with 3 cops. Let P, Q denote the two paths of length 3 obtained when removing r_1, r_2 from T_2 and consider a search $C_1, \ldots, C_t$ of T'_2. We also set $S_1 := \{r_1, s_1\}$ and $S_2 := \{r_2, s_2\}$. First, note that if each set C_i intersects both S_1 and S_2, then the robber has a winning strategy to evade the cops by choosing any of the two paths P, Q and staying in it the whole time (such a path will be occupied by at most one cop at each step, allowing the robber to win). Assume now that this is not the case, and let $i \in [t]$ be the first step of time for which one of the two sets S_1, S_2, say S_1 does not intersect C_i. If $i = 1$, then the robber immediately wins by choosing s_1 as its initial position, so we assume that $i \geqslant 2$. Then for each $1 \leqslant j < i$, both paths P and Q are occupied by at most one cop, and moreover one of them, say P is occupied by no cop at step $i-1$ (if $i \geqslant 2$). In particular, as $|C_{i-1} \Delta C_i| \leqslant 1$ and as S_1 contains a vertex from $C_{i-1} \setminus C_i$, P is also not occupied by a cop at step i. A winning strategy for the robber then consists in staying in P during the first $i-1$ steps and escaping so far the only cop that enters P, and then reaching s_1 at step i. We assume from now on that $k \geqslant 3$, and that Items 1 and 2 holds in T_{k-1}.

In what follows, we let H_1, H_2 denote the two disjoint copies of T_{k-1} in T_k, and u_i^j ($i, j \in [2]$) denote the copies of r_1, r_2 in H_1, H_2, as depicted in Section VII.6. We moreover let H'_1 and H'_2 denote the two disjoint copies of T'_{k-1} in T_k, such that for each $i \in [2]$, H'_i contains H_i, and for each $i, j \in [2]$, we let v_i^j denote the copies of the vertices s_1, s_2 in H'_1, H'_2, as depicted in Section VII.6. We moreover set $W_j := \{v_1^j, v_2^j\}$ for each $j \in \{1, 2\}$.

We start proving Item 1 and consider a search $C_1, \ldots, C_t$ on G with k

[6] Recall that in order to move, the cop must first disappear from the graph. This follows from the condition $|C_{i-1} \Delta C_i| \leqslant 1$ in the definition of a search.

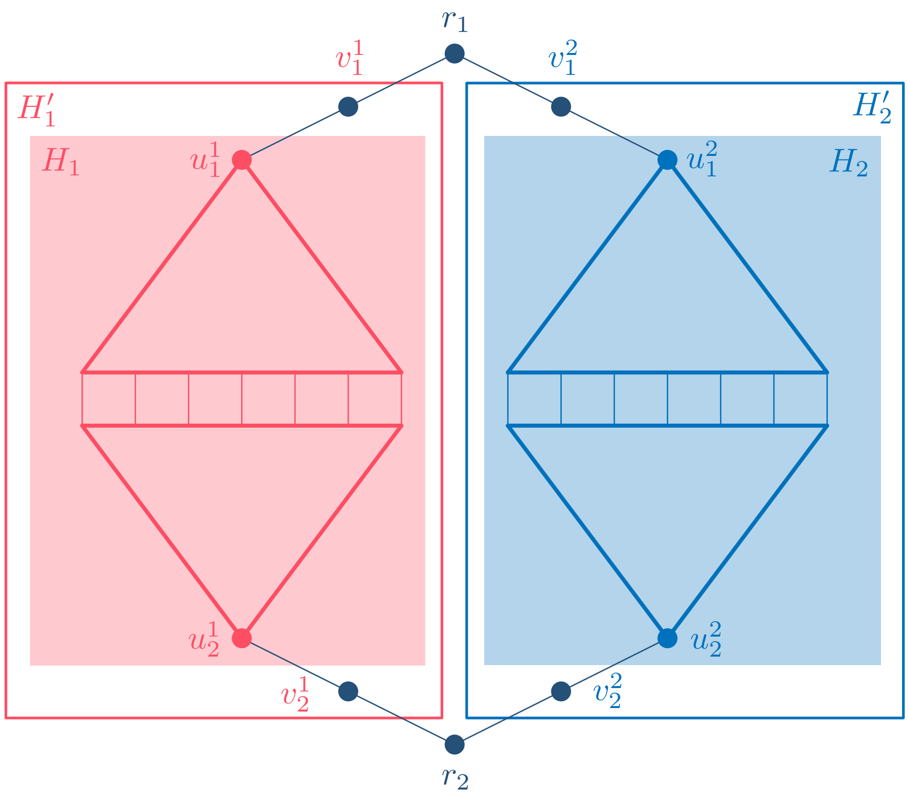


Figure VII.22: Schematic description of the decomposition of T_k in the proof of Lemma VII.6.2. The triangles represent copies of the complete binary trees of height $k-2$ involved in the definition of T_{k-1}.

cops. We will show that this search is evasive. Intuitively, the strategy simply consists for the robber in playing the W_j-node search game in some subgraph H_j' ($j \in [2]$). Applying Item 2 from induction hypothesis on H_j', the robber will be able either to escape the cops until the end of the search, or to jump into H_{3-j}' by travelling through some vertex of the set W_j, and then start playing iteratively the W_{3-j}-node search game in H_{3-j}'. This is slightly more technical to prove formally, as one must ensure that when jumping from H_j' to H_{3-j}', the robber can moreover reach some initial configuration of the winning strategy he will use later in H_{3-j}'.

More formally, for each $i \in [t]$, we construct the *projection* $\pi_1(C_i) \in \binom{V(H_1')}{\leqslant k}$ of C_i on H_1' as follows. First, for each $u \in C_i \cap V(H_1')$, we add u to $\pi_1(C_i)$. Then, if C_i contains any of the vertices r_1, v_1^2, u_1^2 (these correspond to the vertices on the shortest $v_1^1u_1^2$-path), we add v_1^1 to $\pi_1(C_i)$. Symmetrically, if C_i contains any of the vertices r_2, v_2^2, u_2^2, we add v_2^1 to $\pi_1(C_i)$. If necessary, for each vertex u in $C_i \cap H_2 \setminus \{u_1^2, u_2^2\}$, we add at least one of the two vertices v_1^1, v_2^1, by respecting the following rule that if H_2 contains at least two vertices from C_i, then both v_1^1 and v_2^1 are added to $\pi_1(C_i)$. We claim that a simple induction on $i \in [t]$ implies that we can construct such sets $\pi_1(C_i)$ with the additional property that $\pi_1(C_1), \dots, \pi_1(C_t)$ is a search on H_1' (with at most k cops). We define symmetrically a projected search $\pi_2(C_1), \dots, \pi_2(C_t)$ on H_2'.

We now describe a winning strategy for the robber to show that $C_1, \dots, C_t$ is evasive. We will ensure that at each step, the robber occupies a vertex of $H_1 \cup H_2$. A *safe state* is a configuration of the game where at some step $i \in [t]$, the robber occupies some vertex in some subgraph H_j ($j \in [2]$) such that $|C_i \cap V(H_j)| \leqslant 1$. We will first show that we may assume that the robber can reach some safe state. We assume without loss of generality that $|V(H_1) \cap C_1| \leqslant k-1$. By induction hypothesis, Item 2 holds in H'_1, implying that the robber has a strategy to win the W_1-node searching game in H'_1 with respect to the projected search $\pi_1(C_1), \ldots, \pi_1(C_t)$. It then implies that one of the following holds:

- either the robber can escape the cops with respect to $\pi_1(C_1), \ldots, \pi_1(C_t)$, in which case he also has a strategy to escape the cops in T_k with respect to the initial search $C_1, \ldots, C_t$, and staying in H'_1 at every step,
- or there exists some step $i \in [t]$ such that the robber can stay in H'_1 and escape to the cops until step i, and then reach one of the two vertices of W_1. Without loss of generality, assume that this vertex is v_1^1. The definition of π_1 then implies in particular that $|C_i \cap V(H_2)| \leqslant 1$, and moreover that none of the vertices v_1^1, r_1, v_1^2, u_1^2 belongs to C_i. This then implies that the robber can safely travel to the vertex u_1^2 at step i, and reach a safe state.

We now prove that if at some step $i \in [t]$, the robber could reach some safe state in which he occupies a vertex x of H_j, then it can either escape the cops until the end of the search by staying in H_j, or reach another safe state at some step $i' > i$. Observe that applying iteratively this claim immediately concludes the proof of Item 1. To prove the claim, the arguments are almost the same as above: assume that at step i, the robber occupies a vertex of H_1 and that $|V(H_1) \cap C_i| \leqslant 1$. If $i = t$, then we immediately conclude. Otherwise, consider the search $\pi_1(C_{i+1}) \cup \{v_1^1, v_2^1\}, \pi_1(C_{i+1}), \ldots, \pi_1(C_t)$ of H'_1 (note that this indeed defines a search as $\pi_1(C_{i+1})$ contains at least one of the vertices v_1^1, v_2^1). By induction hypothesis, the robber has a winning strategy for this search when playing the W_1-node searching game in H'_1, for which its initial position is some vertex $y \in V(H_1)$. We now observe that as H_1 is 2-connected, and as $|V(H_1) \cap C_i| \leqslant 1$, the robber can, at the end of step i, move from x to y without being caught by the cops. He can then reproduce the winning strategy in H'_1 for the W_1-node searching game, and according to the outcome, the same arguments used in the previous case distinction imply that he can either escape the cops until the last step, or reach another safe state for some $i' > i$.

We now show Item 2, using Item 1 from the induction hypothesis. The proof is basically the same as for the case $k = 2$. Let $C_1, \ldots, C_t$ be a search with $k+1$ cops on T'_k. We let $W := \{s_1, s_2\}$, $S_1 := \{v_1^1, r_1, s_1, v_1^2\}$ and $S_2 := \{v_2^1, r_2, s_2, v_2^2\}$, so that H_1, H_2 are the two connected component of $G - (S_1 \cup S_2)$. If for each $i \in [t]$, $|C_i \cap V(H_1)| \leqslant k-1$, the robber can escape to the cops by reproducing some winning strategy given by Item 1 in H_1, implying that $C_1, \ldots, C_t$ is evasive. The same applies symmetrically if $|C_i \cap V(H_2)| \leqslant k-1$ for each $i \in [t]$. Otherwise, we can consider the minimum $i \in [t]$ such that $|C_i \cap V(H_1)| = k$, and we may moreover assume that $|C_{i'} \cap V(H_2)| \leqslant k-1$ for each $i' \leqslant i$. In particular,

C_i must be disjoint to at least one of the sets S_1, S_2. If $i = 1$, then the robber immediately wins the W-node search game by choosing s_1 as its initial position. Otherwise, by induction hypothesis, Item 1 holds in H_2 and implies that the robber has a winning strategy to escape the cops until step i, while staying in H_2. In particular, as $|C_i \cap V(H_1)| = k$, we then have $|C_i \cap V(H_2)| \leqslant 1$. As H_2 is 2-connected, it then implies that at the start of step i, the robber can safely reach one of the sets S_1, S_2 which is disjoint from C_i, and thus also W. This concludes the proof of Item 2.

□

VII.7 Conclusion

As mentioned in the introduction, the best lower bound we know on the possible pathwidth of graphs edge-coverable by k shortest paths is k. Together with Theorem VII.1.4, it suggests that the upper bound from Theorem VII.1.2 could be potentially improved to a linear bound.

Question VII.7.1. *Let G be a graph edge-coverable by k shortest paths. Is it true that $\mathrm{pw}(G) = O(k)$? And that $\mathrm{pw}(G) \leqslant k$?*

Another observation going in the direction of Question VII.7.1 is that graphs edge/vertex-coverable by k shortest paths have *cop number* at most k [AF84]. In particular, it is well-known (and not hard to prove) that the cop number of a graph is upper bounded by its treewidth, and thus also by its pathwidth.

The second question which is left open by our work (and asked in [DFPT24]) concerns the existence of a polynomial upper-bound on the pathwidth of graphs which are vertex-coverable by k shortest paths. At the moment, it does not seem for us that the ideas from Section VII.3 generalize to this case.

Another natural question consists in asking if Theorem VII.1.5 generalizes in some way to larger values of k:

Question VII.7.2. *Does there exist a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $k \geqslant 3$ and every graph G edge-coverable by k isometric subtrees, we have $\mathrm{tw}(G) \leqslant f(k)$? If yes, can f be polynomial?*

However, Bastide, Duron, Hodor, Liu and Nie [BDH+25] found constructions of graphs edge-coverable by 4 isometric subtrees that contain arbitrarily large subdivided walls as a subgraph, implying a negative answer to Question VII.7.2. A question left open by their work concerns the existence of graphs edge-coverable by 3 isometric subtrees and with arbitrarily large treewidth. It might also be interesting to find some restricted hypothesis under which Question VII.7.2 could hold.

References

[AF84] Martin Aigner and Michael Fromme. "A game of cops and robbers". English. In: *Discrete Applied Mathematics* 8 (1984), pp. 1–11. ISSN: 0166-218X. DOI: 10.1016/0166-218X(84)90073-8.

[BBG+17] Taylor Ball et al. "On the cop number of generalized Petersen graphs". English. In: *Discrete Mathematics* 340.6 (2017), pp. 1381–1388. ISSN: 0012-365X. DOI: 10.1016/j.disc.2016.10.004.

[BDH+25] Paul Bastide et al. *Counterexamples to statements on isometric graph coverings*. 2025.

[BK17] Kristóf Bérczi and Yusuke Kobayashi. "The directed disjoint shortest paths problem". English. In: *25th European symposium on algorithms, ESA 2017, Vienna, Austria, September 4–6, 2017. Proceedings*. Id/No 13. Wadern: Schloss Dagstuhl – Leibniz Zentrum für Informatik, 2017, p. 13. ISBN: 978-3-95977-049-1. DOI: 10.4230/LIPIcs.ESA.2017.13.

[BMG+26] Julien Baste et al. "A Polynomial Bound on the Pathwidth of Graphs Edge-Coverable by k Shortest Paths". In: *43rd International Symposium on Theoretical Aspects of Computer Science, STACS 2026, Grenoble, France, March 9-13, 2026*. Ed. by Meena Mahajan et al. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026, 10:1–10:17. DOI: 10.4230/LIPICS.STACS.2026.10. URL: https://doi.org/10.4230/LIPIcs.STACS.2026.10.

[BNRZ23] Matthias Bentert et al. "Using a Geometric Lens to Find k-Disjoint Shortest Paths". In: *SIAM Journal on Discrete Mathematics* 37.3 (2023), pp. 1674–1703. DOI: 10.1137/22M1527398. eprint: https://doi.org/10.1137/22M1527398. URL: https://doi.org/10.1137/22M1527398.

[CCFV26] Dibyayan Chakraborty et al. "Isometric path complexity of graphs". In: *Discrete Mathematics* 349.2 (2026), p. 114743. ISSN: 0012-365X. DOI: https://doi.org/10.1016/j.disc.2025.114743. URL: https://www.sciencedirect.com/science/article/pii/S0012365X25003516.

[CDD+22] Dibyayan Chakraborty et al. "Complexity and Algorithms for ISOMETRIC PATH COVER on Chordal Graphs and Beyond". In: *33rd International Symposium on Algorithms and Computation (ISAAC 2022)*. Ed. by Sang Won Bae and Heejin Park. Vol. 248. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 12:1–12:17. ISBN: 978-3-95977-258-7. DOI: 10.4230/LIPIcs.ISAAC.

2022.12. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ISAAC.2022.12.

[CDF+25] Dibyayan Chakraborty et al. *Parameterized complexity of isometric path partition: treewidth and diameter*. Preprint, arXiv:2508.05448 [cs.DS] (2025). 2025. URL: https://arxiv.org/abs/2508.05448.

[Cou90] Bruno Courcelle. "The monadic second-order logic of graphs. I: Recognizable sets of finite graphs". English. In: *Information and Computation* 85.1 (1990), pp. 12–75. ISSN: 0890-5401. DOI: 10.1016/0890-5401(90)90043-H.

[DFPT24] Maël Dumas et al. "On graphs coverable by k shortest paths". English. In: *SIAM Journal on Discrete Mathematics* 38.2 (2024), pp. 1840–1862. ISSN: 0895-4801. DOI: 10.1137/23M1564511.

[Eil98] Tali Eilam-Tzoreff. "The disjoint shortest paths problem". In: *Discrete Applied Mathematics* 85.2 (1998), pp. 113–138. ISSN: 0166-218X. DOI: https://doi.org/10.1016/S0166-218X(97)00121-2. URL: https://www.sciencedirect.com/science/article/pii/S0166218X97001212.

[FF01] David Fisher and Shannon Fitzpatrick. "The isometric path number of a graph". In: *JCMCC. The Journal of Combinatorial Mathematics and Combinatorial Computing* 38 (Jan. 2001).

[FHW80] Steven Fortune, John Hopcroft, and James Wyllie. "The directed subgraph homeomorphism problem". English. In: *Theoretical Computer Science* 10 (1980), pp. 111–121. ISSN: 0304-3975. DOI: 10.1016/0304-3975(80)90009-2.

[HP25] Meike Hatzel and Michał Pilipczuk. *On graphs coverable by chubby shortest paths*. 2025.

[Kin92] Nancy G. Kinnersley. "The vertex separation number of a graph equals its path-width". English. In: *Information Processing Letters* 42.6 (1992), pp. 345–350. ISSN: 0020-0190. DOI: 10.1016/0020-0190(92)90234-M.

[KKR12] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. "The disjoint paths problem in quadratic time". In: *Journal of Combinatorial Theory, Series B* 102.2 (2012), pp. 424–435. ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2011.07.004. URL: https://www.sciencedirect.com/science/article/pii/S0095895611000712.

[KP85] Lefteris M. Kirousis and Christos H. Papadimitriou. "Interval graphs and searching". In: *Discrete Mathematics* 55.2 (1985), pp. 181–184. ISSN: 0012-365X. DOI: https://doi.org/10.1016/0012-365X(85)90046-9. URL: https://www.sciencedirect.com/science/article/pii/0012365X85900469.

[KU16] Kolja Knauer and Torsten Ueckerdt. "Three ways to cover a graph". English. In: *Discrete Mathematics* 339.2 (2016), pp. 745–758. ISSN: 0012-365X. DOI: 10.1016/j.disc.2015.10.023.

[Loc21] Willian Lochet. "A Polynomial Time Algorithm for the k-Disjoint Shortest Paths Problem". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2021), pp. 169–178. DOI: 10.1137/1.9781611976465.12. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611976465.12. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611976465.12.

[RS95] N. Robertson and P.D. Seymour. "Graph Minors .XIII. The Disjoint Paths Problem". In: *Journal of Combinatorial Theory, Series B* 63.1 (1995), pp. 65–110. ISSN: 0095-8956. DOI: https://doi.org/10.1006/jctb.1995.1006. URL: https://www.sciencedirect.com/science/article/pii/S0095895685710064.

Chapter VIII

MDS when forbidding a minor of pathwidth 2

Abstract

We show that graphs excluding a minor H of pathwidth 2 admit a deterministic $f(H)$-round 50-approximation distributed algorithm for Minimum Dominating Set (MDS). We build upon the results of Chapter IV that handles the subcase of $H = K_{2,t}$. Until then, though fast and approximate distributed algorithms for such problems were already known for H-minor-free graphs, all of them had an approximation ratio depending on the size of H. This points toward a deeper truth behind the behavior of MDS in H-minor-free graphs. Indeed, we conjecture that by allowing the number of rounds to be an arbitrary function of H, we can obtain an approximation ratio within constant factor of the *pathwidth* of H. A matching lower bound guarantees that it would be optimal if true. Our algorithm is rather elementary, and the analysis, while intricate, is self-contained except for a theorem on the asymptotic dimension of minor-closed classes.

Acknowledgements

This section is part of a work with Marthe Bonamy, Cyril Gavoille, Tim Planken, and Alexandra Wesolek, adapted to the thesis. The work of this chapter is not yet published.

Contents

VIII.1 Introduction

The aim of chapter section is to extend the results of Chapter IV, to larger families, and to show that all H-minor-free graphs have a constant-factor approximation in the LOCAL model when $\mathrm{pw}(H) = 2$.

Noting that $K_{2,t}$ has pathwidth 2, for $t \geqslant 2$, we propose the following strong generalization of Theorem IV.3.1.

Theorem VIII.1.1. *For every graph H of pathwidth at most 2, there is a $_H(1)$-round 50-approximation deterministic LOCAL algorithm for Minimum Dominating Set on H-minor-free graphs.*

This is a first step in the direction of Conjecture V.1.2.

Conjecture V.1.2. *For every graph H, there is an $O(\mathrm{pw}(H))$-approximation $O_H(1)$-round LOCAL algorithm for Minimum Dominating Set on H-minor-free graphs.*

For this, we use the same Algorithm 5 as on $K_{2,t}$-minor-free graphs, but with further analysis to show that the same algorithm still works for any H with pathwidth 2. Notice that we can still bound the number of vertices in local 1-cuts and of interesting vertices in local 2-cuts by Lemmas IV.2.1 and IV.2.2, as those lemmas apply for any class of graphs of bounded asymptotic dimension (and minor-free graphs have asymptotic dimension bounded by 2 [BBE+24, Proposition 1.17]). Thus, we just need to prove an equivalent of Lemma IV.3.2 for H-minor-free graphs when $\mathrm{pw}(H) = 2$. We cannot rely on Ding's characterization [Din17] anymore, and we are left with studying the structure of such graphs ourselves.

The first difficulty in proving this is that there are many graphs of pathwidth 2, and we would rather not tailor the proof to each of them. To circumvent this, we use the notion of universal major.

Universal major for graphs of pathwidth 2. We first construct a graph which is universal under minor containment for graphs of pathwidth at most 2 and of bounded size.

The graph L_k, the ladder with k railings, is defined as in Section VIII.1.

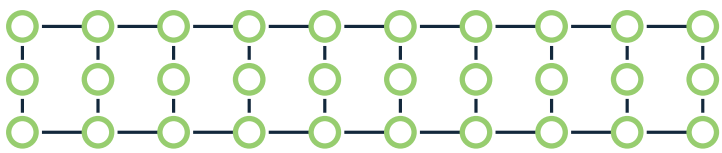


Figure VIII.1: The graph L_k, here for $k = 10$. It is constructed from a $k \times 2$-grid, a ladder, where the k vertical edges have been subdivided into two edges.

The corners of the subdivided grids are the *ends* of the ladder. Any set of two vertices of L_k where the vertices are at distance 2, of degree 2, and adjacent to a common vertex of degree 2 is an *end* of L_k. A *railing* is a vertex of degree 2 that is not an end of L_k. The *paths* of the ladder are the two paths that are obtained by removing every railing.

We first use the following property.

Proposition VIII.1.2. *Every graph H of pathwidth 2 and with k vertices is a minor of L_k.*

Proof. Let G be a graph of pathwidth 2, let (P, β) be a path decomposition of G of width 2 and let $k = |V(G)|$. We prove by induction on $|P|$ that for any subset S of size 2 in the last bag of P, G is contained in L_k as a minor with the additional condition that S is rooted in one end of L_k, i.e. for the two vertices of S, their corresponding branch set each contain one vertex of the end. The base case where $|P| = 1$ is easy to check. Let $P = P'yx$ and $B = \beta(x) = \{u, v, w\}$ be the last bag of P. Without loss of generality, we can assume that $\beta(x) \neq \beta(y)$, and therefore we can also assume that $\beta(x) \cap \beta(y) \subseteq \{u, v\}$ and that $G[\beta(x)]$ is a clique. There are two cases: either $S = \{u, v\}$ or $S = \{u, w\}$ (the case $S = \{v, w\}$ is similar). Let us handle the two cases.

- In the first case where $S = \{u, v\}$, as S is a cut of G and separates w from the rest of the graph, we can apply the induction hypothesis to $G' = G - w$ and S. We get a ladder L_{k-1} that contains G' as a minor with end S. Now, it is simple to show that L_k contains G as a minor. Let $\{u', v'\}$ be an end of L_{k-1} such that u' (respectively v') is contained in the vertex branch set of L_{k-1} of u (respectively v). Add a path $u'u''w'v''v'$ of length 5 at the end of the L_{k-1}, and map u'' (resp. v'') to the same branch set as u' (resp. v'), and map w' to w. As the branch set corresponding to u contains u' (and similarly for v), we are done.

- In the other case where $S = \{u, w\}$, we can use a similar reasoning. Indeed, we can get an L_{k-1} as before by induction, such that $\{u, v\}$ is rooted at one of its ends. Then extend as before this L_{k-1} to an L_k by adding a path $u'u''w'v''v'$ of length 5 at its ends. Map u'' to the branch set of u, and w', v' to the branch set of w. This creates a ladder L_k which contains G as a minor and such that S rooted in $\{u'', v''\}$.

This finishes the proof. □

The goal of the rest of the chapter is to prove the following lemma.

Theorem VIII.1.3. *Let k be an integer, and let G be an L_k-minor-free and true-twin-free graph. Let X be the set of minimal r-local 1-cuts, I the set of interesting vertices inside minimal r-local 2-cuts. Finally, let $Y = X \cup I$ and let G''_r be the graph obtained from G by deleting any edge internal to $N[Y]$ and by deleting any isolated vertex in the resulting graph. Then every connected component of G''_r has weak diameter bounded by $d_{VIII.1.3}(k)$.*

In the rest of this chapter, we assume that the diameter of some connected component of G''_r is more than $d_{VIII.1.3}(k)$, and we will derive a contradiction. We can restrict our attention to the case where the interesting condition on the vertices of 2-cuts is dropped. This reduction is integrated in the proof of Theorem VIII.1.3. The main intuition is that non-interesting vertices are adjacent to components of small diameter, as they are dominated by a single vertex, so we can mostly ignore them.

Our main contribution is to point towards an interesting behavior for Minimum Dominating Set in H-minor-free graphs, and to give some partial evidence for it. The obvious question is now: what stops us from proving Theorem VIII.1.1 for H with pathwidth 3 (or larger)? This is discussed in Chapter IX.

Definitions. A set X is an r-local cut if $G[\bigcup_{u \in X} N^r[u]]$ is a connected graph and X is a cut in it. We are specifically interested in *minimal* r-local cuts, which are r-local cuts where every vertex has neighbors in at least two connected components of $G[\bigcup_{u \in X} N^r[u]] \setminus X$. Equivalently, they are the r-local cuts whose proper subsets are either not r-local cuts or create fewer connected components.

An r-local cut is an r-local k-cut if it has size k. Note that testing whether a set of size k is an r-local k-cut can be done in $O(kr)$ rounds in the local model.

Organization of this chapter. We begin by developing the tools needed to prove Theorem VIII.1.3. In Section VIII.2, we establish a local analogue of Menger's theorem. We then apply this result in Section VIII.3 to construct very large subgraphs, called almost-ladders, in a less general special case. In Section VIII.4, we show how very large almost-ladders can be transformed into large ladders. Finally, Section VIII.5 explains how to reduce the general setting to the simpler case considered in Section VIII.3, and combines all the ingredients to complete the proof of Theorem VIII.1.3.

VIII.2 Local Menger

The goal of this section is to prove the following local version of Menger's theorem. It states that if some vertices u and v are connected by some path P, either u and v are connected by k internally disjoint paths that live inside the local neighborhood of P, there exists an r-local cut of size strictly less than k intersecting P.

Theorem VIII.2.1. *Given some k, some $r \geqslant 1$, some graph G, and any two vertices $u, v \in V(G)$ along with any uv-path P, one of the following holds.*

- *Either there are k disjoint uv-paths in $G[N^{2r+1}[P]]$, or*
- *there is a minimal r-local $(< k)$-cut involving a vertex of P.*

Proof. We proceed by induction on k then by induction on $|P|$ (note that we may consider P to be induced). The base case for $k = 1$ is trivial, as P

is itself a uv-path. The base case for $|P| = 2$ follows from the induction hypothesis on k. Indeed, consider the graph $G' := G - uv$ obtained from G by removing the single edge of P. Then G' does not contain any r-local $(< k-1)$-cut, as if there would be such a cut C, there would be an r-local $(< k)$-cut $C \cup \{u\}$ in G (note that distances can only decrease in G'). Therefore, one can obtain $k-1$ disjoint uv-paths in G' and add the edge uv to create the kth path between u and v. This proves the base case.

Now, suppose that $|P| \geqslant 3$. Let w be the neighbor of u in P, and note that for P' such that $P = uP'$, we still have $|P'| \geqslant 2$. Assume that there is no r-local $(< k)$-cut involving a vertex of P (hence none involving a vertex of P'). By induction on $|P|$, there are k disjoint wv-paths $Q_1, \ldots, Q_k$ in $G[N^{2r+1}[P']]$.

We are interested in the property $(H_i)_{1 \leqslant i \leqslant k}$ as follows:

(H_i) There is a collection of k pairwise internally vertex-disjoint paths in $G[N^{2r+1}[P]]$ such that i of them are uv-paths (not containing w) and $k-i$ of which are wv-paths (not containing u).

We first observe that (H_1) holds. Since w does not r-separate u from the other neighbor of w (different from u) in P', there is a path R_1 from u to $Q_1, \ldots, Q_k$ in $G[N^{r+2}[u]]$. In particular, let j be the first index j such that R_1 intersects Q_j. By considering the path obtaining by following R_1 until it intersects Q_j then following Q_j to v, and keeping the other $(Q_\ell)_{\ell \neq j}$ as before, we obtain the desired collection.

We now argue that (H_i) implies (H_{i+1}) if $1 \leqslant i < k$. Note that (H_{k-1}) implies the desired property: one can take the path $Q_1 \cup \{u\}$ (extending Q_1 with the edge uw) along with the paths $R_1, R_2, \ldots, R_{k-1}$.

Let $R_1, \ldots R_i$ and $Q_1, \ldots, Q_{k-i}$ be a collection of (respectively uv- and wv-) paths as given by (H_i). Let A be the set of vertices that are internal to some R_j, adding u itself. Finally, let B be the set of vertices that are internal to some Q_j, adding v itself. Note that $w \notin A \cup B$. Consider the edges involved in some R_j as *blue* and orient the blue edges toward v in the natural way. We say a directed walk W in $G[N^{2r+1}[P]]$ is a *good walk* if it starts in $A \cap N^r[u]$ and ends in B, visits vertices not in $A \setminus \{u\}$ at most once (and does not visit w at all), does not visit B except for its final vertex, does not repeat edges, any blue edge in W appears backward, any vertex in $A \setminus \{u\}$ is preceded or followed (or both) by a blue edge.

Claim VIII.2.2. *If there is a good walk W starting in u, then (H_{i+1}) holds.*

Proof of claim. We proceed by induction on the number of blue edges in W. If W does not contain any blue edge, then it does not intersect $A \setminus \{u\}$ and is in fact a path. Let j be such that W ends in Q_j (see Section VIII.2), and let Q'_j be the suffix that goes from the endpoint of W to v. We note that R_{i+1} can be defined as $W \cup Q'_j$, and the collection of (Q_ℓ)'s re-arranged to delete Q_j. This yields a family of paths in $N^{2r+1}[P]$, as W is a good walk and so $W \subseteq N^{2r+1}[P]$.

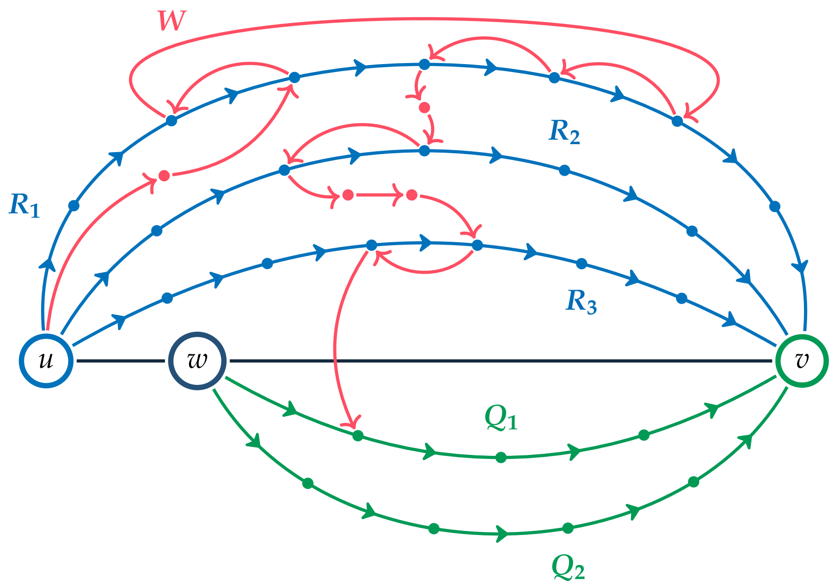


Figure VIII.2: A good walk W in red, starting at u. Vertices of edges in blue are in A and the ones in green are in B.

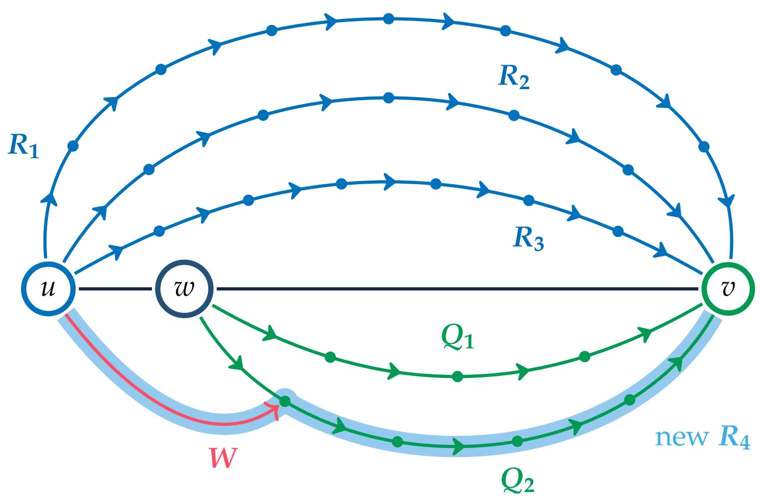


Figure VIII.3: The base case of the induction for $j = 2$.

If W contains a blue edge, we build an additional path starting from u. Let j be the first Q_j visited by W. We restrict ourselves to the graph $G' = G[R_1 \cup \ldots \cup R_i \cup Q_j]$. We claim that there exists $i+1$ disjoint uv-paths. If it is not the case, by Menger's theorem, there must exist some cut S of size at most i separating u from v in G'. Because the paths $R_1, \ldots, R_i$ exist in G', each path $R_{j'}$ contains a unique vertex $s_{j'}$ of S. Let A^- be the set of vertices of A that appear in some $R_{j'}$ before the vertex $s_{j'}$, excluding this vertex. Let A^+ be the set of vertices of A that appear in some $R_{j'}$ after the vertex $s_{j'}$, including this vertex. The walk W contains a path P' from $R_1 \cup \ldots \cup R_i$ to Q_j, with its internal vertices outside of Q_j. Therefore, this path must have one of its endpoints in Q_j and the other in A^+ (it could not be in A^-, as S would not be a cut). Let $x \in P_{j'}$ be this endpoint vertex. As W is a good walk, it intersects A^- (in u) and then intersects $A^+ \setminus S$ (at a vertex after $s_{j'}$ on path $R_{j'}$). Therefore, there must exist a path from A^- to A^+ internally disjoint from A, contradicting the fact that S is a cut. This gives $i+1$ paths disjoint from all the $Q_{j'}$ for $j' \neq j$, and proves (H_i). For an illustration of the result, see Section VIII.2. $\diamond$

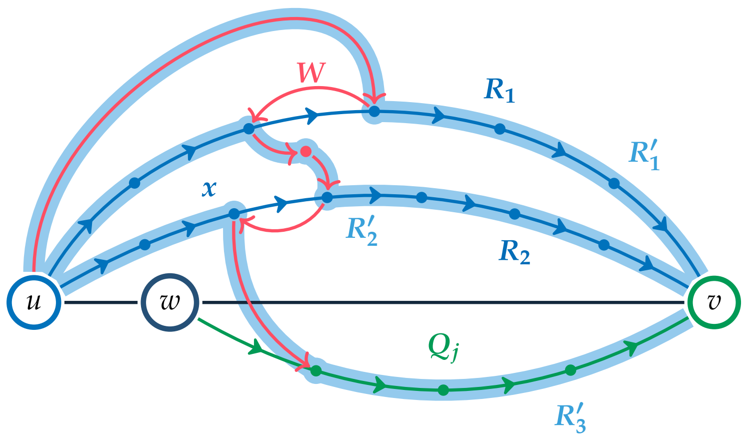


Figure VIII.4: An illustration of the induction step. W is represented in red.

Claim VIII.2.3. *There is a good walk starting in u.*

Proof of claim. Let $X \subseteq \{1, \ldots, i\}$ be the set of every $j \in X$ such that there is a good walk starting from R_j. As a reminder, a good walk has to start from $N^r[u]$, therefore $j \in X$ if some good walk starts from $R_j \cap N^r[u]$. For each $j \in X$, we let r_j be the vertex of R_j closest to u in R_j (possibly $r_j = u$). If $r_j = u$ for some j then the conclusion holds. We aim to argue that if $r_j \neq u$ for every $j \in X$ then some good walk starts from some $R_{j'}$ with $j' \notin X$, a contradiction.

We let A_1 be the set of vertices in A which strictly predate r_j on their R_j, adding u itself, and A_2 the set of vertices in A which come strictly after r_j on their R_j. Note that $A_1 \cup A_2$ is in $N^r[u]$ and does not intersect any R_ℓ for $\ell \notin X$. By assumption, there is no good walk starting in A_1.

Since $i \leqslant k-2$, the set $Y = \{w\} \cup \{r_j \mid j \in X\}$ has size at most $k-1$. Note that $Y \subseteq N^r[u]$ as every r_j is in A_1. Moreover, notice that $A_1 \cap N^r[Y]$ is non-empty because it contains u. $(A_2 \cup B) \cap N^r[Y]$ is also non-empty because it contains the vertex of Q_1 closest to w. We claim that Y is not an r-local cut. Indeed, if it was, let Y' be a minimal r-local cut contained entirely in Y. We would have $w \in Y'$ because w is adjacent to two vertices in different components: $u \in A_1$ and $b \in B$ where b is the vertex adjacent to w in Q_1 (so removing w from Y' would decrease the number of components). Therefore, there is a path R' between A_1 and $A_2 \cup B$ in $G[N^r[Y] \setminus Y]$. It follows that $R' \subseteq N^r[Y] \subseteq N^{2r+1}[u]$.

Shorten R' (by removing from it some prefix path) such that it starts in $x \in A \setminus A_2$ and does not contain any other vertex in $A \setminus A_2$. Let j be such that $x \in R_j$. Either $j \notin X$ or $j \in X$ (and in this case, $x \in A_1$). Either way, if R' does not contain any vertex in A_2, the path R' is a good walk starting from x, a contradiction with either the definition of A_1 or the definition of X. Thus, let y be the first vertex of R' that belongs to A_2, and let ℓ be such that $y \in R_\ell$. Let W_ℓ be a good walk starting from r_ℓ, and let W' (see Section VIII.2) be the walk obtained by concatenating the prefix of R' ending in y with the subpath of $\overleftarrow{R}_\ell$ from y to r_ℓ then with W_ℓ. As W_ℓ is a good walk, $\overleftarrow{R}_\ell$ is a subpath of R_ℓ, and $R' \subseteq N^{2r+1}[u]$, it follows that $W' \subseteq N^{2r+1}[P]$. Note that the walk obtained from W' by removing any closed subwalk is a good walk starting from x, a contradiction with the fact that either $x \in A_1$ or $j \notin X$.

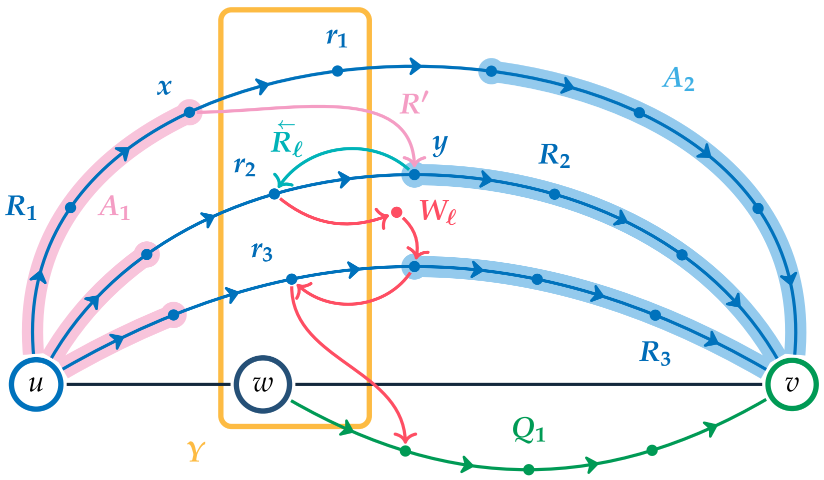


Figure VIII.5: An illustration of the proof with $j = 1 \in X$ and $\ell = 2$.

$\diamond$

Claims VIII.2.2 and VIII.2.3 together imply the wanted statement. $\square$

VIII.3 *Building almost-ladders*

In this section, we build large subgraphs that almost look like ladders We convert those large subgraphs into ladders later on in Section VIII.4. We also work on a simpler case, where all 2-cut vertices are taken by the algorithm, and not only interesting 2-cut vertices.

Let u, v be vertices in a graph G. Let P be a path with start-vertex u and end-vertex v. We call a P *minimal* in G if every uv-path P' such that

$V(P') \subseteq V(P)$ satisfies $P' = P$. This is equivalent to saying that whenever two vertices $x, y \in P$ are adjacent in G then they are consecutive on P. Note that any subpath of a minimal path is minimal.

The *end-distance* $d_G(P)$ of P in G is the distance of u and v in G, i.e. $d_G(P) := d_G(u,v)$. Moreover, we say that a vertex $x \in V(G)$ *(ρ,r)-separates* P if $N_G^\rho[x]$ separates u from v in $G[N_G^r[P]]$. We say that P is a *(ρ,r)-loop* if it has end-distance $\leqslant r$ and there is a vertex $x \in P$ with distance $\geqslant \rho$ to both u and v. Moreover, we say that P *contains a (ρ,r)-loop* if it contains a subpath that is a (ρ,r)-loop.

Lemma VIII.3.1. *Let $\rho, r \in \mathbb{N}$ with $\rho \geqslant r$. Let G be a graph and let P be a path in G. Assume that P does not contain a $(\rho, 2r+1)$-loop. Then every vertex $x \in P$ $(\rho+r, r)$-separates P.*

Proof. Fix some $x \in V(P)$ and assume for sake of contradiction that x does not $(\rho+r,r)$-separate P. Then there exists a uv-path $Q \subseteq G[N_G^r[P]] - N_G^{\rho+r}[x]$. For each $y \in V(Q)$, since $y \in N_G^r[P]$, there must exist a vertex $p(y) \in V(P)$ with $d_G(y,p(y)) \leqslant r$. Because $y \notin N_G^{\rho+r}[x]$, we have $p(y) \notin N_G^\rho[x]$. Let P_u be the subpath of P from u to x, and P_v be the subpath of P from x to v.

We now cover Q by two sets $A = \{y \in V(Q) \mid d_G(y, P_u \setminus N_G^\rho[x]) \leqslant r\}$ and $B = \{y \in V(Q) \mid d_G(y, P_v \setminus N_G^\rho[x]) \leqslant r\}$. Since $V(Q) \subseteq A \cup B$, there must be either a vertex of Q in $A \cap B$ or an edge of Q with endpoints in A and B respectively. Either way, there exist vertices $a \in P_u - N_G^\rho[x]$ and $b \in P_v - N_G^\rho[x]$ such that $d_G(a,b) \leqslant 2r+1$.

Let us consider the subpath L of P composed of the vertices in-between a and b. As $a \in P_u$ and $b \in P_v$, the vertex x lies in L. Moreover, $a, b \notin N_G^\rho[x]$, so x is at distance more than ρ from both a and b. Therefore, L is a $(\rho, 2r+1)$-loop, a contradiction. □

Lemma VIII.3.2. *Let $\rho, \ell \in \mathbb{N}$ and let $d \geqslant d(\rho,\ell) := (2\rho+2r+1)\ell + 1$. Let G be a graph and let P be a path of end-distance d in G. Then at least one of the following holds:*

(1) *there are $x_1, x_2, \ldots, x_\ell \in P$ such that every x_i $(\rho+r,r)$-separates P, and such that the balls $N_G^{\rho+r}[x_i]$ are pairwise disjoint and contain neither u nor v;*

(2) *P contains a $(\rho, 2r+1)$-loop.*

Proof. Assume that (2) does not hold, and let us show (1). By applying Lemma VIII.3.1, we get that every vertex of P $(\rho+r,r)$-separates P. It remains to find ℓ vertices that are pairwise at distance at least $2\rho+2r+1$ (so that their distance-$(\rho+r)$ neighborhoods are disjoint), and at distance at least $\rho+r+1$ from u and v. Consider the function $f : x \in V(P) \mapsto d_G(u,x)$. For any two adjacent vertices x and y, the values $f(x)$ and $f(y)$ differ by at most 1. As $f(u) = 0$ and $f(v) = d$, every integer between 0 and d is in $f(P)$.

Therefore, for every $i = 1, \ldots, \ell$, one can take the value $t_i = \rho + r + 1 + (i-1)(2\rho + 2r + 1)$ and consider at vertex x_i of P such that $f(x_i) = t_i$. Let us check that the x_i's meet the required conditions.

- For every $i = 1, \ldots, \ell$, we have $t_i \geqslant t_1 = \rho + r + 1$, i.e. $d_G(u, x_i) \geqslant \rho + r + 1$.
- For every $i = 1, \ldots, \ell$, we have $t_i \leqslant t_\ell = \rho + r + 1 + (\ell - 1)(2\rho + 2r + 1) = d(\rho, \ell) - \rho - r - 1$, i.e. $d_G(x_i, v) \geqslant \rho + r + 1$.
- For every $1 \leqslant i < j \leqslant \ell$, we have $t_j - t_i = (2\rho + 2r + 1)(j - i) \geqslant 2\rho + 1$, i.e. $d_G(x_i, x_j) \geqslant 2\rho + 2r + 1$.

This finishes the proof. □

Let G be a graph. An *almost-ladder with k rungs* is a subgraph of G composed of two disjoint paths P_1 and P_2, and pairwise disjoint paths $R_1, R_2, \ldots, R_k$ such that every R_i is a P_1P_2-path in G. The paths P_j are called *rails*, and the paths R_i are called *rungs*. Note that the interior of rung R_i is allowed to intersect the rail P_j. If the interiors of the rungs are disjoint from the rails and all non-empty, then the almost-ladder is *strong*.

Lemma VIII.3.3. *Let $r, \ell, k, d \in \mathbb{N}$ with $d \geqslant d_{VIII.3.3}(r, \ell, k)$. Let G be a graph and let P be a path in G of end-distance $\geqslant d$. Then at least one of the following holds:*

(a) There is $\rho \in \mathbb{N}$, a subpath $P' \subseteq P$ and vertices $x_1, x_2, \ldots, x_\ell$ that $(\rho + r, r)$-separate P' and are such that the balls $N_G^{\rho+r}[x_i]$ are pairwise disjoint and contain neither u nor v;

(b) G contains an almost-ladder with k rungs such that each rung has length at most $2r + 1$ and such that the rails are contained in P.

Moreover, the sequence $(d_{VIII.3.3}(r, \ell, k))_k$ is defined recursively so that $d_{VIII.3.3}(r, \ell, 0) = 0$ and $d_{VIII.3.3}(r, \ell, k+1) = (2d_{VIII.3.3}(r, \ell, k) + 10r + 7)\ell + 1$ for any integer $k \geqslant 1$.

By doing standard computations, one obtains

$$d_{VIII.3.3}(r, \ell, k) = ((10r + 7)\ell + 1) \cdot \frac{(2\ell)^k - 1}{2\ell - 1} = O\left(r \cdot (2\ell)^k\right).$$

Proof. We prove this statement by induction on k. Note that for $k = 0$, (b) is true by vacuity. Now, suppose that the property holds for k; let us show that the lemma holds for $k + 1$.

The path P has end-distance at least

$$d_{VIII.3.3}(r, \ell, k+1) = (2(d_{VIII.3.3}(r, \ell, k) + 4r + 3) + 2r + 1)\ell + 1 .$$

Choose $\rho := d_{VIII.3.3}(r, \ell, k) + 4r + 3$ and assume that (a) is not satisfied for this value of ρ. By Lemma VIII.3.2, P contains a $(\rho, 2r + 1)$-loop L. Let a and b be the ends of L. By definition of L, there exists a vertex x

that is at distance at least ρ from both a and b. Consider the connected components of $L - N^{4r+2}[a]$. One must contain x and a vertex y at distance $4r + 3$ from a. Let Q be the subpath of this connected component, consisting of vertices in-between x and y. The path Q has length at least $\rho - 4r - 3 = d(r, \ell, k)$, and is disjoint from $N^{4r+2}[a]$. (a) cannot hold for Q, as otherwise it would hold for P.

Therefore, there exists an almost ladder with k rungs $R_1, \dots, R_k$ each of length at most $2r + 1$ and with rails P_1, P_2 that are subpaths of Q. We order the vertices of L from a to b. Without loss of generality, all vertices of P_1 appear before the vertices of P_2 on Q. Said differently, the vertices of P_1 are closer to a on the path P than any vertex of P_2 is. We construct a subpath P_1' of L by starting with P_1 and adding all vertices of L that are between P_1 and a. Similarly, construct the subpath P_2' of L consisting of all vertices of L between P_2 and b. It follows that P_1' and P_2' are subpaths of P which are disjoint. There is no P_1'–P_2' edge in G, as P is minimal. Moreover, all R_i's are $P_1'P_2'$-paths in G.

As L is a $(\rho, 2r + 1)$-loop, the vertices a and b are at distance at most $2r + 1$, i.e. there is a path R_{k+1} from a to b of length at most $2r + 1$. Notice that this path is entirely contained in $N^{2r+1}[a]$ by definition, and that all rungs are contained in $N^{2r+1}[V(P_1) \cup V(P_2)]$ by definition. The path Q does not intersect $N^{4r+2}[a]$. In particular, R_{k+1} is a $P_1'P_2'$-path in G disjoint from all other R_i's.

This proves that P_1', P_2' along with $R_1, \dots, R_k, R_{k+1}$ form an almost-ladder with $k + 1$ rungs, each of length at most $2r + 1$. This proves (b), and finishes the proof. □

Let r and k be integers. Let G be a graph, and let B_r be the set of vertices involved in a minimal r-local $(\leqslant 2)$-cut in G. We let G_r' be the graph obtained from G by deleting any edge internal to $N[B_r]$ and by deleting any isolated vertex in the resulting graph.

Next, we show that outcome (a) of Lemma VIII.3.3 leads to the existence of an almost-ladder, in the case where P is entirely contained in a connected component G_r'. For this, we also need to use Theorem VIII.2.1.

Lemma VIII.3.4. *Let $r, \ell \in \mathbb{N}$. Fix some shortest path P of a connected component of G_r', with ends u and v. Suppose that there is a $\rho \in \mathbb{N}$, a subpath $P' \subseteq P$ and vertices $x_1, x_2, \dots, x_\ell$ that $(\rho + 2r, 2r)$-separate P' and are such that the balls $N_G^{\rho+2r}[x_i]$ are pairwise disjoint and do contain neither u nor v. Then G contains a strong almost-ladder with $\lceil \ell/3 \rceil$ rungs.*

Proof. Define the vertex sets $B_i = N^{\rho+2r}[x_i]$ for any $1 \leqslant i \leqslant \ell$, which are all pairwise disjoint. We apply Theorem VIII.2.1 to u, v and P. Let Q_1, Q_2 and Q_3 be the resulting paths. Note that those paths are entirely contained in $G[N^{2r}[P]]$, and therefore are separated by the B_i's. In particular, all B_i's intersect all Q_j's.

Now, fix some $1 \leqslant i \leqslant \ell$. We claim that there exist distinct $a_i, b_i \in \{1, 2, 3\}$ and a path $R_i \in G[B_i]$ such that the ends of R_i are in Q_{a_i} and Q_{b_i} respectively, and such that the interior of R_i is non-empty and disjoint from $Q_{a_i} \cup Q_{b_i}$. As $G[B_i]$ is connected and B_i intersects all the paths Q_1, Q_2 and Q_3, one of the following holds.

- There is a path in $G[B_i]$ intersecting all three paths Q_1, Q_2 and Q_3. Choose such a path R_i that has minimum length while still maintaining the intersection property. Let a_i, b_i such that R_i intersects Q_{a_i} and Q_{b_i} at its ends x and y, respectively. First, $a_i \neq b_i$, as x and y cannot be in the same Q_j; this would contradict minimality. Suppose for sake of contradiction that some interior vertex of R_i is in Q_{a_i}, let z be the last one of R_i (the closest y in R_i). Consider R'_i, the subpath of R_i composed of the vertices between z and y. If this subpath intersects Q_1, Q_2 and Q_3, we are done. Otherwise, it contains only vertices from Q_{a_i} and Q_{b_i}. Additionally, the interior vertices of R'_i are only in Q_{b_i}, otherwise this would contradict the minimality of R_i. Let c_i be the only element of $\{1,2,3\} \setminus \{a_i, b_i\}$. Q_{c_i} intersects $R_i - R'_i$, i.e. the subpath of R_i composed of the vertices from x to z contains an interior vertex z' in Q_{c_i}. One can take $z' \in Q_{c_i} \cap R_i$ closest to y on R_i. The subpath of R_i composed of vertices from z' to y intersects all three paths, contradicting the minimality of R_i. Therefore, we get that the interior of R_i cannot intersect Q_{a_i}. The same reasoning applies for Q_{b_i} too, so the interior of R_i is disjoint from $Q_{a_i} \cup Q_{b_i}$. The interior of R_i is non-empty, as R_i intersects all of Q_1, Q_2 and Q_3.

- Assume now that no path in $G[B_i]$ intersects all three of Q_1, Q_2 and Q_3. Since B_i is connected and intersects Q_1, Q_2 and Q_3, there exists a subgraph T intersecting those three paths. Choose T minimal with respect to inclusion. Then T is a tree and not a path, and because it is minimal, every leaf is either in Q_1, Q_2 or Q_3. They also cannot be in the same Q_j. Therefore, there are three leaves $\ell_1, \ell_2, \ell_3 \in V(T)$ that are respectively in Q_1, Q_2, Q_3. As T is a tree with three leaves, there exists a unique vertex x of degree 3. For $j \in \{1,2,3\}$, let x_j be the vertex closest to x on T that intersects $Q_1 \cup Q_2 \cup Q_3$. We claim that $x_j \in Q_j$ for every j (and therefore $x_j \neq x$, and more specifically $x_j = \ell_j$). Suppose for sake of contradiction that it is not the case, that is, $x_j \in Q_c$ for $c \neq j$. Let d be the third index, so that $\{c, d, j\} = \{1,2,3\}$. Then the path in T from ℓ_j to ℓ_d contains a vertex from every path Q_1, Q_2, Q_3, a contradiction. Now, take R_i as the unique path of T joining x_1 to x_2 (that is, we choose $a_i = 1$ and $b_i = 2$). The interior of R_i is non-empty, as it contains the degree 3 vertex, and is disjoint from $Q_1 \cup Q_2$, by definition of x_1 and x_2.

In both cases, the claim is proven.

Each i thus determines a pair $\pi(i) \in \{\{1,2\}, \{1,3\}, \{2,3\}\}$ such that R_i is a $Q_a Q_b$-path with $\pi(i) = \{a, b\}$. By the pigeonhole principle, there exists a pair $\{a, b\}$ and a set I of size at least $\ell/3$ such that $\pi(i) = \{a, b\}$ for every $i \in I$. Define P_1 as the path $Q_a - \{u, v\}$ and the path P_2 as the path $Q_b - \{u, v\}$. P_1 and P_2 are internally vertex disjoint, and for each $i \in I$, the path R_i is a $P_1 P_2$-path, with interior disjoint from $P_1 \cup P_2$. The R_i's are pairwise disjoint, because they are contained in pairwise disjoint balls. This proves that P_1, P_2 along with $(R_i)_{i\in I}$ form an almost-ladder with $\lceil \ell/3 \rceil$ rungs. Moreover, it is strong because the rungs all have interior non-empty and disjoint from $Q_a \cup Q_b \subseteq P_1 \cup P_2$, and are disjoint from the rails. □

VIII.4 Building ladders from almost-ladders

We are now ready to construct our ladder from our slightly larger almost-ladder obtained by Lemma VIII.3.3.

We first transform the almost-ladders obtained by outcome (b) of Lemma VIII.3.3 into strong almost-ladders.

Lemma VIII.4.1. *Let H be an almost-ladder with k rungs in a graph G, where the rails are subpaths of some minimal path. Then G contains a strong almost-ladder with k rungs.*

Proof. Let H be an almost-ladder with rails P_1 and P_2, and with rungs $R_1, R_2, \ldots, R_k$. We first show that there is an almost-ladder $H' \subseteq H$ with k rungs such that the interior of each rung is disjoint from $P_1 \cup P_2$. Since every rung R_i is a P_1P_2-path, there is a subpath $R'_i \subseteq R_i$ whose interior is disjoint from $P_1 \cup P_2$ (e.g. take the subpath of R_i starting at the last vertex that lies on P_1 and ends on the first vertex thereafter on P_2). Then P_1 and P_2 together with the rungs $R'_1, R'_2, \ldots, R'_k$ builds the desired strong almost-ladder H'. Indeed, every R'_i has non-empty interior, as P is a minimal path. □

Then, we show that strong almost-ladders can be turned into slightly smaller ladders.

Lemma VIII.4.2. *Suppose G contains a strong almost-ladder with $(k-1)^2+1$ rungs. Then G contains a ladder with k rungs. In particular, L_k is a minor of G.*

Proof. Let H be an almost-ladder with rails P_1 and P_2, and with rungs $R_1, R_2, \ldots, R_k$ such that the interior of each R_i is disjoint from $P_1 \cup P_2$ and non-empty. Denote the start-vertex of rung R_i by s_i, and the end-vertex by t_i. Note that the t_i's and s_i's are pairwise distinct since the rungs R_i are pairwise disjoint. Without loss of generality, assume that the order of the s_i's on P_1 is $s_1, s_2, \ldots, s_k$. By the Erdős-Szekeres Theorem, we find $t_{i_1}, t_{i_1}, \ldots, t_{i_k}$ such that they either appear in this order or in the reverse order on the path P_2. Then, P_1, P_2 together with $R_{i_1}, R_{i_2}, \ldots, R_{i_k}$ is a ladder with k rungs. □

Lemmas VIII.4.1 to VIII.3.4 together imply the following theorem.

Theorem VIII.4.3. *Let $r, k \in \mathbb{N}$. Then, for all $d \geqslant d_{VIII.4.3}(r, k)$ such that for any graph G, if some connected component C of G'_r has weak diameter at least d, then G contains L_k as a minor.*
Moreover, $d_{VIII.4.3}(r, k) := d_{VIII.3.3}(2r, 3((k-1)^2+1), (k-1)^2+1)$.

Proof. Let u and v be two vertices of C at distance d, and let P be a shortest uv-path in $G[C]$. In particular, P is minimal with end-distance $d \geqslant d_{VIII.3.3}(2r, 3((k-1)^2+1), (k-1)^2+1)$. Apply Lemma VIII.3.3 to P, with parameters $r_{VIII.3.3} := 2r$, $\ell_{VIII.3.3} := 3((k-1)^2+1)$ and $k_{VIII.3.3} := (k-1)^2+1$. If outcome (a) holds, then by Lemma VIII.3.4, G contains a strong almost-ladder with $(k-1)^2+1$ rungs. If outcome (b) holds, then by Lemma VIII.4.1, G contains a strong almost-ladder with $(k-1)^2+1$ rungs. In either case, Lemma VIII.4.2 finishes the proof. □

VIII.5 Finishing the proof

In G, let X be the set of minimal r-local 1-cuts, I the set of interesting vertices inside minimal r-local 2-cuts. Let C_1 be the set of r-local 1-cuts and C_2 the set of r-local 2-cuts. Finally, let $Y = X \cup I$ and let G''_r be the graph obtained from G by deleting any edge internal to $N[Y]$ and by deleting any isolated vertex in the resulting graph.

Theorem VIII.5.1. *Let $r \geqslant 2$, and let $D \in \mathbb{N}$. Let G be a true-twin-free graph. Suppose that every connected component of G'_r has weak diameter at most D. Then every connected component of G''_r has weak diameter at most $D+4$.*

Proof. I is the set of interesting vertices with the original definition. Let $\widehat{I}$ be the set obtained by dropping the condition $N[v] \not\subseteq N[u]$. Thus, $v \in \widehat{I}$ if there is a minimal r-local 2-cut $\{u, v\}$ such that at least two connected components of $G[N^r_G[\{u,v\}]] - \{u, v\}$ contain a vertex non-adjacent to u.

Claim VIII.5.2. *As G has no true twins, we have $N_G[I] = N_G[\widehat{I}]$.*

Proof of claim. Clearly $I \subseteq \widehat{I}$, so $N_G[I] \subseteq N_G[\widehat{I}]$. Conversely, let $v \in \widehat{I} \setminus I$. Then there exists a minimal r-local 2-cut $\{u, v\}$ such that at least two connected components of $G[N^r_G[\{u,v\}]] - \{u, v\}$ contain a vertex non-adjacent to u. Since $v \notin I$, we have $N_G[v] \subseteq N_G[u]$. As G has no true twins, this containment is proper, i.e. $N_G[v] \not\supseteq N_G[u]$.

Every vertex non-adjacent to u is also non-adjacent to v. Therefore, the same two local components also contain vertices non-adjacent to v. Thus, the same cut $\{u, v\}$, with the roles of u and v reversed, witnesses that $u \in I$.

But $N_G[v] \subseteq N_G[u]$, and $u \in I$. Hence, every vertex of $N_G[v]$ already is in $N_G[I]$. Therefore, we get $N_G[\widehat{I}] \subseteq N_G[I]$, and even $N_G[I] = N_G[\widehat{I}]$. ◇

Therefore, for the rest of the proof, we can drop the condition of neighborhood containment (i.e. $N_G[v] \nsubseteq N_G[u]$) when working on interesting vertices.

We also need this claim, which is the only place where we use $r \geqslant 2$.

Claim VIII.5.3. *Let $c = \{u, v\}$ be a minimal r-local 2-cut, and suppose that $v \notin Y$. Then, in $G - c$, at most one connected component contains a vertex non-adjacent to u.*

Proof of claim. Suppose not. Then there are two distinct connected components K_1, K_2 of $G - c$ and vertices $z_i \in V(K_i)$, $i = 1, 2$, such that z_i is non-adjacent to u.

For each i, since K_i is a component separated by the cut c, it contains a vertex adjacent to u or to v. Choose a path in K_i from such a vertex to z_i, and let z_i' be the first vertex on this path which is non-adjacent to u. If z_i' is the first vertex of the path, it is adjacent to v, and $d_G(z_i', c) \leqslant 1$. Otherwise, the predecessor of z_i' on the path is adjacent to u, and so $d_G(z_i', u) \leqslant 2$. In both cases, since $r \geqslant 2$, we have $z_i' \in N_G^r[c]$.

Moreover, z_1' and z_2' are in distinct connected components of $G[N_G^r[c]] - c$, because they already are in distinct connected components of $G - c$. Each of these two local components contains a vertex non-adjacent to u. As we dropped the $N[v] \nsubseteq N[u]$ condition, then v is in I, a contradiction. This proves the claim. ◇

Now let C be a connected component of G_r''. First, we compare C with the components of G_r'. Recall that B_r is set of vertices involved in minimal r-local 1-cuts or minimal r-local 2-cuts, so that G_r' is obtained by deleting the edges internal to $N_G[B_r]$ and then deleting isolated vertices.

Since $Y \subseteq B_r$, the graph G_r' is obtained from G_r'' by deleting additional edges, namely those coming from minimal r-local 2-cuts whose vertices are not in Y.

Fix such a remaining minimal r-local 2-cut $c = \{u, v\}$, with $v \notin I$. By the claim, all but at most one component of $G - c$ consist entirely of vertices adjacent to u. Call those components u-dominated.

If K is a u-dominated component of $G - c$, then every vertex of K lies in $N_G[u]$. Also, every edge incident with a vertex of K has its other endpoint in $K \cup c$, because K is a connected component of $G - c$. Hence, every such edge is internal to $N_G[B_r]$. Therefore, all edges incident with K are deleted when constructing G_r', and the vertices of K become isolated and are deleted. Thus, no u-dominated component contributes a vertex to G_r'.

We get that for every remaining minimal r-local 2-cut, at most one of its associated connected components can contain vertices of G_r'. It follows that the component C of G_r'' intersects at most one connected component of G_r'. Indeed, if two distinct components of G_r' intersected C, then a path in C between them would have to pass through some remaining minimal r-local 2-cut and use two different sides of that cut which both survive in G_r', a contradiction.

Let C^* be the unique component of G'_r met by C, if such a component exists. If no such component exists, then C is made only of vertices lying in dominated components of remaining non-interesting 2-cuts, together with the corresponding cut vertices. In that case C has weak radius at most 2, and hence weak diameter at most $4 \leqslant D + 4$. So assume from now on that C^* exists.

We claim that every vertex of C has distance at most 2 from C^* in G. Let $x \in V(C)$. If $x \in V(C^*)$, we are done. Otherwise, x is deleted when constructing G'_r from G''_r. Hence, x is in one of the u-dominated components described above, for some remaining minimal r-local 2-cut $\{u, v\}$. Thus, x is adjacent to u. Since C is connected and C^* is the only surviving component of G'_r inside C, the cut vertex u has a neighbor in C^*, on the unique surviving side which contains C^*. We $d_G(u, C^*) \leqslant 1$ and hence $d_G(x, C^*) \leqslant 2$.

Since C^* is a connected component of G'_r, its weak diameter is at most D. It follows that C has weak diameter at most $D + 4$. This proves the theorem. □

Theorems VIII.4.3 and VIII.5.1 together imply Theorem VIII.1.3.

As said before, we use the same Algorithm 5 as on $K_{2,t}$-minor-free graphs, but with further analysis to show that the same algorithm still works for any H with pathwidth 2. We can still bound the number of vertices in local 1-cuts and of interesting vertices in local 2-cuts by Lemmas IV.2.1 and IV.2.2, as those lemmas apply for any class of graphs of bounded asymptotic dimension (and minor-free graphs have asymptotic dimension bounded by 2 [BBE+24, Proposition 1.17]). As we have proven Theorem VIII.1.3, an equivalent of Lemma IV.3.2 for L_k-minor-free graphs. By Proposition VIII.1.2, this lemma applies to all H-minor-free graphs with $\mathrm{pw}(H) = 2$. Thus, this proves Theorem VIII.1.1.

References

[BBE+24] Marthe Bonamy et al. "Asymptotic Dimension of Minor-Closed Families and Assouad–Nagata Dimension of Surfaces". In: *Journal of the European Mathematical Society* 26.10 (2024), pp. 3739–3791. DOI: 10.4171/JEMS/1341.

[Din17] Guoli Ding. *Graphs without large $K_{2,n}$-minors*. 2017. DOI: 10.48550/ARXIV.1702.01355. URL: https://arxiv.org/abs/1702.01355.

Part V

To Pathwidth k, and Beyond!

Chapter IX
MDS when forbidding a minor of arbitrary pathwidth

Abstract

In Chapter V, we conjectured that for every graph H, there is an $O(\mathrm{pw}(H))$-approximation $O_H(1)$-round LOCAL algorithm for Minimum Dominating Set on H-minor-free graphs. We proved the case $\mathrm{pw}(H) = 2$ in Chapter VIII. We survey the challenges that are to be overcome to generalize our algorithms to arbitrary pathwidth.

Acknowledgements

This section is part of a work with Marthe Bonamy, Cyril Gavoille, Tim Planken, and Alexandra Wesolek, adapted to the thesis. The work of this chapter is not yet published.

Contents

IX.1 Introduction

Everything Embeddable All at Once. Let us give a quick introduction about the graph minors structure theorem [RS03], to build intuition on what challenges we need to overcome to prove our general conjecture.

The graph minors structure theorem [RS03], proved by Robertson and Seymour, is one of the central results of modern structural graph theory. Roughly speaking, it states that if a graph excludes a fixed graph H as a minor, then it cannot be arbitrarily complicated, and can be described in a combinatorial and topological way. Its global structure must be assembled from pieces that are, in some sense, close to graphs embeddable on a surface that is simple, in the sense that H cannot be embedded on it.

More precisely, the theorem states that every H-minor-free graph is formed of pieces that are almost embeddable in some surface of bounded genus, and then is obtained from those pieces by clique-sums, i.e. an operation that glues two graphs together along a common clique, possibly after deleting some edges inside that clique. Each piece (before taking the clique sums), is embeddable on a surface of bounded Euler genus up to a bounded number of "errors". One must allow a bounded number of exceptional vertices, called *apices*, whose adjacencies can be arbitrary, as well as a bounded number of *vortices*, which are regions of bounded pathwidth attached along the boundary of the surface. Although this description is technically involved, the intuition is clear: excluding a fixed minor forces all complex parts of the graph to be obtained through a bounded number of well-understood obstructions. In the above, all bounding functions are dependent on the size $|V(H)|$ of the minors. Said differently, the Euler genus, the number of apices, the number of vortices, and their respective pathwidth are all bounded by a function of the size of H. Recently, those functions have been shown [GSW25] to be polynomial in $|V(H)|$.

This theorem plays a foundational role because it explains in a structural way how minor-closed graph classes are sparse. It is especially useful to design centralized graph algorithms. Roughly, once a graph is known to be built from nearly embeddable pieces with only a bounded number of irregularities, one can often transfer methods that works on planar graphs and graphs embedded on low-genus surfaces to the much wider world of minor-free graphs. The theorem itself, and the techniques developed to prove this theorem are used countless times in the literature. Although it is rarer, it has also been used for designing distributed algorithms [HLZ18].

The standard method to use this result in algorithms goes as follows. One can analyze each nearly embeddable piece separately, and handle in some way the (bounded number of) apex vertices and vortices. The latter are most often the complex part to handle. Then, one can combine the local solutions through the clique-sum decomposition.

A tentative algorithm. The algorithms we developed for $K_{2,t}$-minor-free graphs and bounded-genus graphs are not isolated results. They both encompass the difficulties encountered when attempting to prove our conjecture for graphs excluding a minor H of arbitrary pathwidth k. For $K_{2,t}$-minor-free graphs, our approach roughly relies on identifying and taking all local separators of size at most 2 in our dominating set. In the context of Robertson and Seymour's structure theorem, this technique effectively takes care of the vertices involved in clique-sums. Indeed, clique-sums naturally form connectivity bottlenecks that are exposed by small cuts. Meanwhile, for the almost embeddable parts of the graph, we can naturally use our bounded-genus algorithm to dominate the planar-like parts.

These two particular techniques suggest the following algorithmic pipeline. First, we select all vertices in r-local cuts of size less than k, and put them into our dominating set. Now, consider the set of remaining undominated vertices, which represent the unresolved portion of our minimum dominating set. Because we have already addressed the clique-sum vertices, these remaining components are structurally much simpler, and well-connected. By the Graph Minors structure theorem, those remaining components should behave like almost embeddable graphs. We should therefore be able to safely run our bounded-genus algorithm on any local neighborhood that has bounded genus.

Another Cross in the Flat Wall. After applying the bounded-genus algorithm, the vertices that remain undominated must belong to small-diameter neighborhoods that are highly non-planar. If these neighborhoods happen to form components of strictly bounded diameter, we can simply gather the entire component's topology into the local views of the nodes and brute-force an optimal solution. Otherwise, these neighborhoods form high diameter regions that are both highly connected (since all small cuts were removed) and topologically complex. Therefore, it is possible to assume our graph contains a large wall (hexagonal grid), which is one of the standard obstructions to low connectivity. According to the Flat Wall Theorem [RS95, Theorem 9.8], a very large minor-free graph containing a very large wall contains a large grid-like structure known as a *flat* wall (up to removing a bounded number of vertices). Essentially, a *flat wall* is a wall that is planar around its center, up to removing components separated from the wall by ($\leqslant 3$)-cuts. However, we already handled the planar-looking neighborhood in a previous step of our algorithm. Thus, the wall we started with cannot be flat, and it must contain macroscopic "crosses" (see Figure IX.1) that break the planarity.

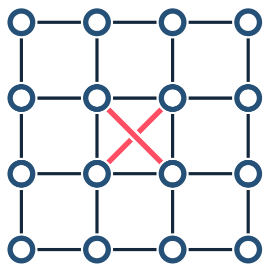


Figure IX.1: An example of a grid with a cross in the middle.

These crosses are families of disjoint paths that route across the wall in a topologically intersecting manner. The presence of these crosses provides a massive amount of options to build large minor-universal graphs. The idea is to build those in within the leftover components, if they are large enough. Specifically, we want to show that these crossing structures can be manipulated to contain as a minor *all* graphs of pathwidth k up to a certain size T.

Crucially, this size threshold T depends only on the radius r of the leftover components, and is entirely independent of k. This allows us to decouple the different parameters. The running time of our distributed algorithm is a function of T, as we must explore far enough to guarantee the existence of this universal minor. On the other hand, the overall approximation ratio is completely independent of T. The brute-force step on bounded-diameter components gives an exact solution (an approximation ratio of 1), and the bounded-genus algorithm provides a constant-factor approximation. Thus, the only part where our approximation ratio can depend on k is the part where we select all vertices in the r-local ($\leqslant k$)-cuts. This separates the parameters: the approximation ratio depends only on the pathwidth bound k of the excluded minor, while the runtime only depends on T.

There is, however, a subtle technicality in formalizing this through the classical graph minors literature. To achieve the desired bounds, we define $k = f(\mathrm{pw}(H))$. Yet, there is currently no version of the Graph Minors structure theorem whose quantitative dependencies rely strictly on the pathwidth of the excluded minor H rather than its total number of vertices $|V(H)|$. Establishing a pathwidth-parameterized structure theorem remains an interesting, open question — if it's even possible to achieve such a feat. Given the monumental complexity of proving even the standard structure theorem, we believe it is significantly easier to prove an ad-hoc version of our distributed approximation theorem that bypasses the heavy machinery of the general structure theorem entirely. To this end, the next section will explore some ideas on how to circumvent the structure theorem by directly creating these large minor-universal structures for graphs of pathwidth k "by hand".

Organization of this chapter. As seen before, our approach boils down to two main challenges: first, bounding the number of vertices contained within the r-local ($\leqslant k$)-cuts. This will be discussed in Section IX.2. Second, bounding the diameter of the highly non-planar components that are left over when executing our algorithms. We elaborate on this in Section IX.3. Each of these challenges require a complicated analysis, and we will discuss them in the next two sections.

IX.2 Avoiding death by a thousand ($\leqslant k$)-cuts

In this section, we discuss how we hope to bound the number of vertices in r-local ($\leqslant k$)-cuts with respect to the size of the minimum dominating set of the graph. As stated, this claim is overly broad and easily shown to be false. Therefore, we must restrict which vertices we bound. We want to bound the number of vertices that are *interesting*. In a graph G, a vertex v is *r-interesting* if there exists some r-local $(\leqslant k)$-cut c containing v such that:

- $N[v] \not\subseteq N[u]$ for any $u \in c \setminus \{v\}$ and
- at least two connected components of $G[N^r[c]] - c$ contain each a vertex non-adjacent to any vertex in $c \setminus \{v\}$.

A $(\leqslant k)$-cut C is *interesting* if it contains an interesting vertex.

To see why the notion of interesting vertices is necessary, consider a large clique with some vertex v. Now, for every vertex $u \neq v$, add a uv-path of 2 edges internally disjoint from the clique. This creates triangles containing the edges incident to v. Consider all sets $\{v, u\}$ where u is a vertex of the original clique (i.e. not of degree 2). Then this set is a 2-cut, and all vertices of the original clique are parts of 2-cuts (but are not interesting). However, the graph we have constructed has a dominating set of size 1 consisting of only v. Therefore, we cannot hope to bound the number of vertices in 2-cuts by a function of $\mathrm{MDS}(G)$.

By using our asymptotic dimension framework developed in Section IV.2.1 for vertices in small cuts (see Section IV.4.1 and Section IV.4.2 for the proofs), we can reduce the problem to bounding the number of interesting vertices in (non-local) $(\leqslant k)$-cuts. For the case $k = 2$, our proof uses the well known SPQR tree decomposition, which essentially decomposes any graph into cycles and 3-connected components. Along with this decomposition tree, there exists a simple characterization of where all possible 2-cuts can lie.

Recently, for $k \in \{3, 4\}$, more involved yet still usable tree-like decompositions have been shown to exist [CK23, KP26]. However, to our knowledge, no such exact decomposition exists for general k. This means that to prove our conjecture, we cannot afford to rely on such decomposition theorems. Instead, we think the right answer is to prove a stronger result through induction.

Conjecture IX.2.1. *For every integer k, there exists some constant c_k such that the following holds. Let D be a dominating set of G, and let $U \subseteq V(G)$ (representing unprocessed vertices). Consider the set I of all vertices in minimal interesting k-cuts that are disjoint from U. Then $|I| \leqslant c_k |D \setminus U|$.*

For $U = \varnothing$, this conjecture directly implies our desired bound. The high-level idea to prove this is by induction on the size of $V(G) \setminus U$, initializing with $U = V(G)$. We consider the set $\widehat{I}$ of all minimal interesting cuts entirely contained within U. We then select a cut $C \in \widehat{I}$ that is closest to U, meaning the connected components of $G - C$ containing U are minimal among all possible choices of C (note that multiple such minimal cuts may exist).

Let A be the union of the connected components of $G - C$ containing vertices of U, and let B be the union of the remaining components, which gives the partition $V(G) = A \sqcup C \sqcup B$. By definition, C separates A from

B. We then classify the remaining cuts in $\widehat{I}$. A cut is *crossing* with respect to C if it intersects multiple components of $G - C$, and otherwise it is *parallel* to C, i.e. it is entirely contained within the neighborhood of a single component of $G - C$.

We want to apply induction with $U' := U \setminus A$. Thus, our goal reduces to bounding the number of vertices in cuts of $\widehat{I}$ that intersect A. By the minimality of C, no such cuts can be parallel to C. Therefore, it suffices to bound only the cuts that cross C. While bounding the number of vertices in these crossing cuts seems to be technically challenging, we believe it is achievable.

IX.3 *Bounding the weak diameter: Around the Component in Bounded Steps*

Universal graphs for bounded pathwidth graphs. Let D' be the set of vertices that are taken by the first steps algorithm, i.e. the union of the interesting vertices in r-local ($\leqslant k$)-cuts and the set taken by the bounded-genus algorithm running in small-distance neighborhoods that are of bounded-genus. In this section, we discuss how we hope to bound the weak diameter of each connected component of $G - D$.

To construct large minor-universal graphs within our non-planar components, it may be useful to look at a particular family of graphs appearing in works of Gorsky, Seweryn, and Wiederrecht [GSW25], which build upon the work of Kawarabayashi, Thomas, and Wollan [KTW18]. In their work, Kawarabayashi et al. [KTW18] defined $H_{w,h}$, a graph obtained from a $(w \times h)$-grid (where h is even) by augmenting the two middle rows with a series of pairs of crossing edges $(i, h/2)(i+1, h/2+1)$ and $(i, h/2+1)(i+1, h/2)$ for every $1 \leqslant i \leqslant w-1$. For an illustration, see Section IX.3. They proved that taking $w, h = t(t-1)$ for $H_{w,h}$ guarantees the existence of a K_t-model within this graph. In Section 4.1 of [GSW25], this is optimized to obtain a smaller bound on the number of required crosses. They leverage the idea that not all crosses are required: one only needs a cross on the two middle rows every third column. Both those proofs can be slightly modified for our purposes, to obtain as a minor from $H_{w,h}$ the graph $P^t_{w'}$ instead of only K_t, and by taking $w = f(w')$ and $h = O(t^2)$.

The significance of constructing a large P^t minor (for long paths P) lies in its universality for bounded pathwidth graphs. Because bounded pathwidth graphs have a path decomposition into sets of size k, they are also clique-sums of sets of bounded size, with the additional property that the underlying clique-sum tree is a path. Therefore, any pathwidth-t graph is contained in a graph formed by gluing K_{t+1}'s along a clique-sum path. Therefore, it is a subgraph of P^{t+1} for a long path P, which proves our claim.

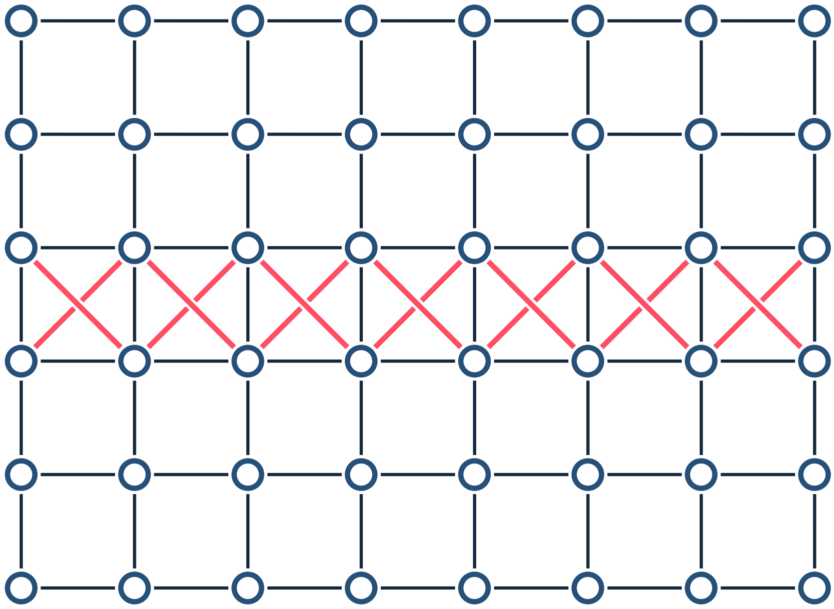


Figure IX.2: An illustration of $H_{w,h}$ (shown here with $w = 8$ and $h = 6$). In addition to the grid, there is a family of mutually intersecting edges (highlighted in red) along the two middle rows $h/2$ and $h/2+1$. These crosses prevent the graph from being embeddable on a surface of bounded genus, even enabling us to construct large minor models such as K_t or P^t for a large path P.

Constructing the grid with crosses. Let us return to our original goal: bounding the diameter of some undominated component C. Given that C is highly connected and inherently non-planar, it is highly likely that they contain large grid-like structures. We believe one could extract a large *wall* where every edge corresponds to a path of strictly bounded length in the original graph. Formally, an $(r \times c)$-wall is a graph obtained from an $(r \times c)$-grid by deleting all vertical edges of the form $(x, y)(x, y+1)$ where $x + y$ is even (or odd, depending on the parity convention). This operation gives a graph of maximum degree 3 that resembles a brick wall.

When reasoning about minor-closed classes, grids and walls are structurally equivalent, as each contains the other as a minor. Specifically, an $(r \times c)$-wall contains an $(\lfloor r/2 \rfloor \times c)$-grid as a minor (by contracting some horizontal edges), and conversely, an $(r \times c)$-grid clearly contains an $(r \times c)$-wall as a subgraph. For our purposes, walls are more practical to work with, because of their maximum degree of 3. Using the property that any graph containing a subcubic graph H as a minor must also contain H as a topological minor (i.e. subdivision), we get that wall minor immediately guarantees the existence of a wall *subdivision*.

Once the wall with short paths is obtained, we can now create crosses within it. Let us give a formal definition of crosses.

Let C be a cycle in a graph G. A C-*cross* in G (see Figure IX.3) is defined as a pair of internally vertex-disjoint paths P_1 and P_2 with endpoints s_1, t_1 and s_2, t_2, respectively, such that these four endpoints appear on C in the alternating order s_1, s_2, t_1, t_2, and the paths are internally disjoint from C.

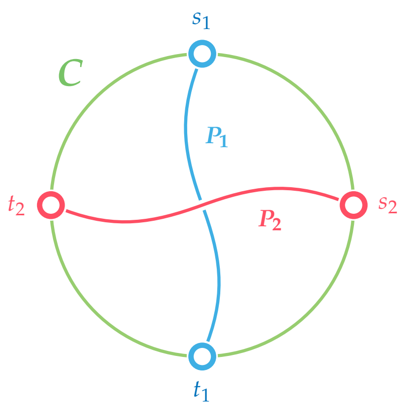


Figure IX.3: An illustration of a C-cross. The cycle C (in green) contains four vertices occurring in the alternating order s_1, s_2, t_1, t_2. The internally disjoint paths P_1 (in blue) and P_2 (in red) connect s_1 to t_1 and s_2 to t_2, respectively.

This concept naturally appears when studying the classic Two Disjoint Paths Problem, which asks whether there exist two vertex-disjoint paths P_1 and P_2 connecting s_1 to t_1 and s_2 to t_2, respectively. We can assume without loss of generality that G contains a cycle C traversing the terminals in the order s_1, s_2, t_1, t_2. Indeed, adding the edges of C to the graph does not alter the feasibility of the disjoint paths. Under this assumption, the Two Disjoint Paths Problem is feasible if and only if G contains a C-cross. Consequently, understanding the broader structural problem of when a graph admits a C-cross provides a complete answer to the disjoint paths problem.

The most immediate obstruction to the existence of a C-cross is topological: if G can be drawn on the plane such that C bounds a face, no C-cross can exist. To see this, add a new vertex inside the face bounded by C and connect it to every vertex of C. The resulting graph is still planar. However, if a C-cross existed, contracting the paths P_1 and P_2 would create a K_5 minor, contradicting planarity.

The other obstruction is because of connectivity bottlenecks. A *separation* in a graph G is a pair (A, B) of vertex subsets such that $A \cup B = V(G)$ and no edge has one endpoint in $A \setminus B$ and the other in $B \setminus A$. The *order* of the separation is exactly $|A \cap B|$. If G admits a separation (A, B) of order at most three such that $V(C) \subseteq A$, the vertices in $B \setminus A$ are useless for forming a C-cross, as one needs to build 4 paths and not 3. We can simplify the graph by deleting $B \setminus A$ and making $A \cap B$ a clique. This forms a new graph H, which is an *elementary C-reduction* of G. If the original graph had a C-cross, this new, smaller graph must also have one. Furthermore, if we choose (A, B) such that some connected component of $G[B \setminus A]$ is adjacent to every vertex in $A \cap B$, the converse is also true. This means applying C-reductions preserves the existence of a C-cross (or lack thereof). We say a graph is a *C-reduction* of G if it can be obtained from G via a sequence of elementary C-reductions.

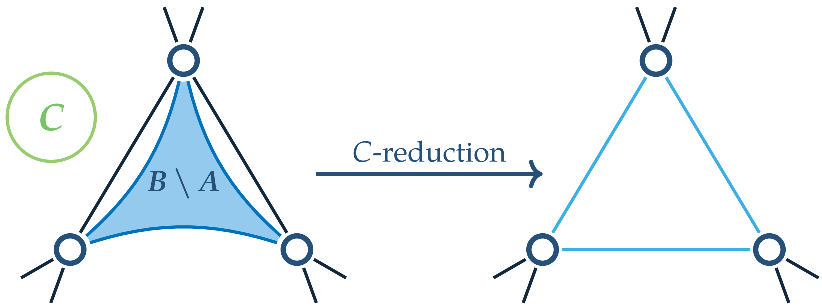


Figure IX.4: An elementary C-reduction. On the left, the blue subgraph $B \setminus A$ is separated from C (in green) by three vertices. During the reduction, the entire internal region is deleted, and the separator vertices are made into a clique (highlighted in blue on the right) to maintain the connectivity of original graph.

Interestingly, once a graph is fully reduced and cannot be C-reducted any further, the only remaining obstruction to finding a C-cross is topological. This theorem, which has appeared multiple times in the literature [Jun70, Sey90, Sey80, Shi80, Tho80], captures this duality.

Theorem IX.3.1 (Two Paths Theorem)**.** *Let G be a graph with some cycle C. Then G does not contain a C-cross if and only if some C-reduction of G can be drawn in the plane such that C bounds a face.*

We now want to construct our crosses. As a reminder, our remaining undominated components contain a large wall where every edge represents a path of bounded length. Within this wall, consider a small cycle C (for instance, the boundary of a single cell or a small cluster of cells). Because the edges of C are short paths in the original graph G, the cycle C itself has a bounded perimeter. Furthermore, we know that the local neighborhood surrounding this wall is fundamentally non-planar.

By Theorem IX.3.1, if the graph cannot be drawn in the plane with C bounding a face, there are only two situations possibles. Either the graph contains a C-cross, or it can be simplified via elementary C-reductions.

Recall that an elementary C-reduction requires a separation (A, B) of order at most 3, i.e. a cut of size at most 3. Because we have already extracted all r-local $(\leqslant k)$-cuts into our dominating set (and assuming $k \geqslant 3$), there are no local cuts of size 3 or less remaining entirely in these components. Therefore, the topological obstruction to planarity manifests purely as a C-cross. With some extra work, we can assume that this C-cross does not intersect too many horizontal, and vertical paths of the wall, thus keeping it somewhat disjoint from the newly created cross.

Because we have assumed, for the sake of contradiction, that the weak diameter of these remaining components is arbitrarily large, the wall itself must have a large weak diameter. This allows us to iteratively apply the same process to distant regions of the wall. Doing so yields a grid with a large number of mutually far-away crosses. From this graph, one can extract a minor similar to $H_{w,h}$. While the dimensions w and h of this extracted grid-with-crosses will be smaller than those of the original wall, they are sufficiently large to generate all possible graphs of pathwidth at most k of a certain size as minors.

Unfortunately, translating all the intuitions into formal proofs is still work in progress.

References

[CK23] Johannes Carmesin and Jan Kurkofka. “Canonical decompositions of 3-connected graphs”. In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 1887–1920.

[GSW25] Maximilian Gorsky, Michał T Seweryn, and Sebastian Wiederrecht. “Polynomial bounds for the graph minor structure theorem”. In: *arXiv preprint arXiv:2504.02532* (2025).

[HLZ18] Bernhard Haeupler, Jason Li, and Goran Zuzic. “Minor excluded network families admit fast distributed algorithms”. In: *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. 2018, pp. 465–474.

[Jun70] Heinz A Jung. "Eine Verallgemeinerung des n-fachen Zusammenhangs für Graphen". In: *Mathematische Annalen* 187.2 (1970), pp. 95–103.

[KP26] Jan Kurkofka and Tim Planken. "A Tutte-type canonical decomposition of 3-and 4-connected graphs". In: *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2026, pp. 2942–3021.

[KTW18] Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. "A new proof of the flat wall theorem". In: *Journal of Combinatorial Theory, Series B* 129 (2018), pp. 204–238.

[RS03] Neil Robertson and Paul D Seymour. "Graph minors. XVI. Excluding a non-planar graph". In: *Journal of Combinatorial Theory, Series B* 89.1 (2003), pp. 43–76.

[RS95] N. Robertson and P.D. Seymour. "Graph Minors .XIII. The Disjoint Paths Problem". In: *Journal of Combinatorial Theory, Series B* 63.1 (1995), pp. 65–110. ISSN: 0095-8956. DOI: https://doi.org/10.1006/jctb.1995.1006. URL: https://www.sciencedirect.com/science/article/pii/S0095895685710064.

[Sey80] Paul D Seymour. "Disjoint paths in graphs". In: *Discrete mathematics* 29.3 (1980), pp. 293–309.

[Sey90] PD Seymour. "Graph minors. IX. Disjoint crossed paths". In: *Journal of Combinatorial Theory Series B* 49.1 (1990), pp. 40–77.

[Shi80] Yossi Shiloach. "A polynomial solution to the undirected two paths problem". In: *Journal of the ACM (JACM)* 27.3 (1980), pp. 445–456.

[Tho80] Carsten Thomassen. "2-Linked Graphs". In: *European Journal of Combinatorics* 1.4 (1980), pp. 371–378. ISSN: 0195-6698. DOI: https://doi.org/10.1016/S0195-6698(80)80039-4. URL: https://www.sciencedirect.com/science/article/pii/S0195669880800394.

Chapter X
MVC and other conjectures

ABSTRACT

We show that in the MINIMUM VERTEX COVER problem, graphs of bounded degeneracy have a constant-factor (independent of the degeneracy) approximation in constant rounds (that depends on the degeneracy) in anonymous networks, a weaker model than the LOCAL model. We also state some conjectures for other problems, such as MINIMUM DISTANCE-k DOMINATING SET, for which we give a constant-factor approximation in constant rounds for bounded-genus graphs. No constant-factor algorithm solving this problem was previously known on planar graphs of arbitrary girth.

ACKNOWLEDGEMENTS

The works of this chapter are not yet published.

Contents

X.1 *Minimum Vertex Cover in degenerate graphs*

X.1.1 *What is the parameter for the approximation factor of MVC?*

We have seen in Conjecture V.1.2 that the right parameter for the approximation ratio of MDS in minor-closed classes might be the pathwidth of the excluded minor. What about other problems? The simplest example of other problems is Minimum Vertex Cover. In this section, we show that the right answer for this problem is simpler: there always is a $(2+\varepsilon)$-approximation algorithm for minor-closed classes, and even for a more general type of class, degenerate graph classes. Thus, unlike in the case of MDS, where the lower bound depends on the pathwidth of the excluded minor, for MVC, the approximation factor remains constant and does not depend on the size of the minor. In addition, our algorithm works in an *anonymous network with a port numbering*. In this model, the input graph is represented by a communication network G whose nodes have no identifiers [Ang80]. Each node $v \in V(G)$ knows only its own degree and has a local numbering of its incident edges by ports $1, 2, \ldots, \deg(v)$ (the *port numbers*). The port numbers at the two endpoints of an edge are not necessarily equal. Communication is done in synchronous rounds. In each round, every node first performs local computation, then sends one message through each of its ports. Then, every node receives one message through each of its ports; one for each neighbor. After T rounds, each node performs local computation and outputs its decision; in our case, whether it joins the returned vertex cover. The number T is the running time of the algorithm. This is a weak model of distributed computing, weaker that the LOCAL model, since the nodes are anonymous.

X.1.2 *Degenerate graphs*

A graph is *k-degenerate* if and only if all of its induced subgraph contains a vertex of degree at most k. Degeneracy is one of the broadest definitions for a sparse class of graphs. In includes bounded expansion classes and minor-free graphs. In hereditary classes, i.e. graph classes closed under taking induced subgraphs, bounded degeneracy is equivalent to bounded average degree.[1] We first show that in a graph of bounded degeneracy, there are not many vertices high degree vertices, with respect to the size of the vertex cover.

[1] To see this, first suppose $\mathscr{C}$ has bounded degeneracy d. Every graph $G \in \mathscr{C}$ can be reduced to the empty graph by removing a vertex of degree at most d, which implies $|E(G)| \leqslant d|V(G)|$. Thus, the average degree of G is $2|E(G)|/|V(G)| \leqslant 2d$, meaning the average degree is bounded. Conversely, suppose every graph in $\mathscr{C}$ has an average degree bounded by some constant c. Let $G \in \mathscr{C}$, and let H be any induced subgraph of G. Because $\mathscr{C}$ is hereditary, $H \in \mathscr{C}$ has average degree at most c. By the handshaking lemma, H must contain at least one vertex whose degree in H is at most c. This proves G is exactly $\lfloor c \rfloor$-degenerate.

Lemma X.1.1. *Let G be a d-degenerate graph, and let X be a minimum vertex cover of G. Then for $D > d$, there are at most $\frac{d}{D-d} \cdot |X|$ vertices of degree at least D in G that are not in X.*

Proof. Let Z denote the set of vertices of degree at least D in G, and set $Y = Z \setminus X$. Since X is a vertex cover, the set $V(G) \setminus X$ is independent, and so in Y. Hence, every edge incident with a vertex of Y has its other endpoint in X.

Now, consider the induced subgraph $G' = G[X \cup Y]$. Each vertex in Y has degree at least D, so there are at least $D|Y|$ edges between Y and X. Since G is d-degenerate, G' has at most $d(|X| + |Y|)$ edges, and $D|Y| \leqslant d(|X| + |Y|)$. Thus, $(D - d)|Y| \leqslant d|X|$, and since $D > d$, this implies $|Y| \leqslant \frac{d}{D-d} \cdot |X|$.

Finally, since $Z \subseteq X \cup Y$, we get

$$|Z| \leqslant |X| + \frac{d}{D-d} \cdot |X| = \frac{D}{D-d} \cdot |X|.$$

This finishes the proof. □

X.1.3 *An algorithm for MVC*

With this, we can design a $(2+\varepsilon)$-approximation in k-degenerate graphs in $f(k)$ rounds. We just need to use any constant-round constant-factor approximation algorithm on bounded degree graphs. Here, we use the 2-approximation algorithm of Åstrand et al. [ÅFP+09].

Theorem X.1.2 ([ÅFP+09])**.** *There exists a 2-approximation algorithm for* MINIMUM VERTEX COVER *on graphs of maximum degree Δ that runs in $(\Delta + 1)^2$ rounds in an anonymous network with a port numbering.*

Note that this algorithm actually uses only 2-bit messages.

Now, we have all the tools necessary to prove our theorem.

Theorem X.1.3. *Let $\varepsilon > 0$. Algorithm 7 is a $(2+\varepsilon)$-approximation for* MINIMUM VERTEX COVER *on d-degenerate graphs that runs in $O\left((d/\varepsilon)^2\right)$ rounds. This algorithm runs in anonymous networks with a port numbering, and uses only 2-bit messages.*

The algorithm goes as follows.

Require: A d-degenerate graph G and a parameter $\varepsilon > 0$.
Ensure: A vertex cover C of G such that $|C| \leqslant (2+\varepsilon) \cdot \mathrm{MVC}(G)$.

1: $Z \leftarrow \{v \in V(G) \mid \deg_G(v) \geqslant (1 + 1/\varepsilon) \cdot d\}$
2: $G' \leftarrow G - Z$
3: Compute a vertex cover C' of G' using the algorithm of Theorem X.1.2 [ÅFP+09].
4: $C \leftarrow Z \cup C'$
5: **return** C

Algorithm 7: A $(2+\varepsilon)$-approximation for MINIMUM VERTEX COVER on d-degenerate graphs.

Proof. Let X be a minimum vertex cover for G, so that $|X| = \mathrm{MVC}(G)$.

First, consider the set of vertices Z added to our cover in Step 1. By Lemma X.1.1, we have

$$|Z \setminus X| \leqslant \frac{d}{(1+1/\varepsilon)\cdot d - d} \cdot \mathrm{MVC}(G) = \varepsilon \cdot \mathrm{MVC}(G).$$

Next, consider the residual graph $G' = G - Z$. Because X is a valid vertex cover for the entire graph G, it follows that $X \setminus Z$ must be a valid vertex cover for G'. Therefore, $\mathrm{MVC}(G') \leqslant |X \setminus Z|$. The 2-approximation algorithm run at Step 3 on G' guarantees that the returned cover C' satisfies $|C'| \leqslant 2|X \setminus Z|$.

Let us now compute the size of $C = Z \cup C'$. Since $Z = (Z \cap X) \cup (Z \setminus X)$, we have

$$\begin{aligned} |C| &= |Z \cap X| + |Z \setminus X| + |C'| \\ &\leqslant |X \cap Z| + \varepsilon \cdot \mathrm{MVC}(G) + 2 \cdot |X \setminus Z| \\ &\leqslant 2 \cdot (|Z \cap X| + |X \setminus Z|) + \varepsilon \cdot \mathrm{MVC}(G) \\ &\leqslant 2 \cdot |X| + \varepsilon \cdot \mathrm{MVC}(G) \leqslant (2+\varepsilon) \cdot \mathrm{MVC}(G). \end{aligned}$$

Moreover, the algorithm works in anonymous networks with port numbering, and uses only 2-bit messages. Indeed, the first step only requires each vertex to inspect its own degree and check whether it's at least $D := (1+1/\epsilon) \cdot d$. Each vertex whose degree is at least D then sends a 1-bit message to all of its neighbors, informing them that it has been selected and should be ignored for the remainder of the algorithm. After receiving these messages, the remaining vertices can simulate the residual graph obtained by deleting the high-degree vertices. This preprocessing takes 1 communication round. The residual graph G' has bounded maximum degree less than $(1+1/\varepsilon) \cdot d$, and the second step applies the deterministic local the algorithm of Theorem X.1.2 [ÅFP+09] to G', which works in the anonymous port-numbering model. As this algorithm takes $(\Delta+1)^2$ rounds on graphs of maximum degree Δ, this step takes $((1+1/\varepsilon) \cdot d + 1)^2$ rounds. Therefore, the whole algorithm runs in at most $((1+1/\varepsilon) \cdot d+1)^2 + 1$ rounds. □

X.2 *Minimum Distance-k Dominating Set on planar graphs*

X.2.1 *A conjecture on the right parameter*

By G^p, we denote the graph obtained from G by adding every edge between vertices at distance at most p in G.

Observation X.2.1. *A solution on G to Minimum Distance-k Dominating Set is a solution on G^k to Minimum Dominating Set.*

Let us recall our lower bound from Chapter V.

For every integer $t \geqslant 1$, let $\mathscr{C}_t$ be the family of graphs composed of the t-th power of a path in which extra vertices of degree one are attached to one extremity of the path.

Lemma V.2.1. *For every t and every function f, every $f(t)$-round deterministic LOCAL algorithm that approximates* Minimum Dominating Set *on $\mathscr{C}_t$ has approximation ratio at least $2t + 1 - \varepsilon$, for every $\varepsilon > 0$.*

Using Observation X.2.1, we immediately get the following corollary.

Corollary X.2.2. *For all integers t, k and every function f, every $f(t)$-round deterministic LOCAL algorithm that approximates* Minimum Dominating Set *on $\mathscr{C}_t$ has approximation ratio at least $2tk + 1 - \varepsilon$, for every $\varepsilon > 0$.*

It is not difficult to see that all the graphs of $\mathscr{C}_t$ have pathwidth at most t. In particular, each graph of $\mathscr{C}_t$ excludes any graph of pathwidth $t+1$ as a minor. Therefore, by choosing $t = \mathrm{pw}(H) - 1$, we get the following analogue of Theorem V.1.1.

Corollary X.2.3. *For every graph H and every $\varepsilon > 0$, there is no $O_H(1)$-round $(2k \cdot \mathrm{pw}(H) - 2k + 1 - \varepsilon)$-approximation deterministic LOCAL algorithm for* Minimum Dominating Set *on H-minor-free graphs.*

Therefore, an immediate question is: is this lower bound optimal? Extending Conjecture V.1.2, we propose the following conjecture.

Conjecture X.2.4. *There exists a function f such that for every graph H, there is an $O(f(k) \cdot \mathrm{pw}(H))$-approximation $O_H(1)$-round LOCAL algorithm for* Minimum Distance-k Dominating Set *on H-minor-free graphs.*

We now present an approximation algorithm for Minimum Distance-k Dominating Set on planar graphs. To do so, we first need to define the "best" distance-r minimum dominating sets of a labelled graph G. Among all dominating sets, we first discard those which contain some v for which there exists w with $N^r[v] \subsetneq N^r[w]$, and among the remaining ones, we declare "best" the one lexicographically smallest considering the labels of the vertices. Let $\mathrm{MDS}_r(G)$ be the size of a minimum distance-r dominating set.

In the rest of this section, we prove the following theorem.

Theorem X.2.5. *Algorithm 8 is an $O(k)$-round $k^{O(k)}$-approximation LOCAL algorithm for* Minimum Distance-k Dominating Set *on planar graphs.*

Note that by using Meta-Theorem I along with Proposition III.7.6, one gets constant-factor approximations on bounded-genus graphs,

Require: A planar graph G with identifiers on the vertices.
Ensure: A distance-r dominating set D of G such that $|D| = r^{O(r)} \cdot \mathrm{MDS}_r(G)$.
1: $D \leftarrow \varnothing$
2: Each vertex u computes $G[N^{5r+1}[u]]$ and all minimum dominating sets of $N^{4r+1}[u]$ in G. Among those sets, u selects the "best" one as D_u.
3: Each vertex u picks the vertex v_u of smallest label in $D_u \cap N^r[u]$, and adds it to D.

Algorithm 8: The algorithm for MINIMUM DISTANCE-k DOMINATING SET on planar graphs.

and more generally on locally-nice classes (see Chapter III for more information).

This partially answers questions of Lenzen and Wenning [LW24] and Akhoondian Amiri and Wiederhake [AW21], who provided constant-factor constant-round approximations for MINIMUM DISTANCE-k DOMINATING SET in bounded expansion classes of high girth, and asked if the high girth assumption can be dropped. By answering this question affirmatively on bounded-genus graphs, we provide a stepping stone for the more general bounded expansion case. Moreover, this proves Conjecture X.2.4 for the particular case of locally-nice graph classes.

X.2.2 Voronoi partitions: introductory definitions

In the next subsections, we describe a way to decompose planar graphs into parts of bounded radius with boundaries of small size. This is necessary in the analysis to prove the approximation factor of the algorithm of Theorem X.2.5.

Voronoi cells and tie-breaking. Let G be a connected planar graph, and let $S \subseteq V(G)$ be a set of vertices called *sites*.

To partition the vertices of G into distinct Voronoi cells without assuming unique distances, we give an arbitrary strict total order $\prec$ on S. We define the *Voronoi cell* of a site $s \in S$, denoted $\mathrm{Vor}(s)$, as the set of vertices $v \in V$ such that for all other sites $s' \neq s$:

- $d(v,s) \leqslant d(v,s')$, and
- if $d(v,s) = d(v,s')$, then $s \prec s'$.

Because every vertex is assigned to exactly one site, the collection $\{\mathrm{Vor}(s)\}_{s \in S}$ forms a partition of $V(G)$.

The dual graph and Voronoi vertices. We now examine the boundaries between these cells. An edge $e = uv \in E(G)$ is defined as a *primal boundary edge* if its endpoints belong to different Voronoi cells, i.e. $u \in \mathrm{Vor}(s_1)$ and $v \in \mathrm{Vor}(s_2)$ with $s_1 \neq s_2$.

Let $G^* = (F, E^*)$ be the planar dual of G, where F is the set of faces of G, and E^* is the set of dual edges. There is a bijection between E and

E^*, such that every edge $e \in E(G)$ is crossed by exactly one dual edge $e^* \in E^*$ connecting the two faces of G incident to e. Let $E^*_B \subseteq E^*$ be the set of dual edges that cross primal boundary edges. We call the graph $G^*_B = (F, E^*_B)$ the *dual boundary subgraph* of G.

In the continuous plane, a Voronoi vertex is a point where three or more Voronoi cells meet. In our graph-theoretic setting, the equivalent of this is a face of G.

A *Voronoi vertex face* is any face $f \in F$ whose boundary contains vertices from three or more distinct Voronoi cells. Equivalently, in the dual boundary subgraph G^*_B, the vertex f is a vertex of degree at least 3.

We define the *abstract dual Voronoi multigraph* $\mathcal{M}$ (also called dual Voronoi diagram in the literature [GKM+21]) as the multigraph obtained from G^*_B by contracting every vertex of degree 2. Therefore, the vertices of $\mathcal{M}$ are the Voronoi vertex faces, and every edge in $\mathcal{M}$ corresponds to a simple path of dual edges in G^*_B where all intermediate faces have degree exactly 2 (meaning they border exactly two cells), and where the endpoints are Voronoi vertices. Said differently, an edge of $\mathcal{M}$ defines a contiguous boundary segment between two faces. Note that the boundary of two Voronoi cells $\mathrm{Vor}(s_1)$ and $\mathrm{Vor}(s_2)$ is not necessarily connected; it can be broken into disjoint parts by a third cell $\mathrm{Vor}(s_3)$ lying in the middle of $\mathrm{Vor}(s_1)$ and $\mathrm{Vor}(s_2)$.

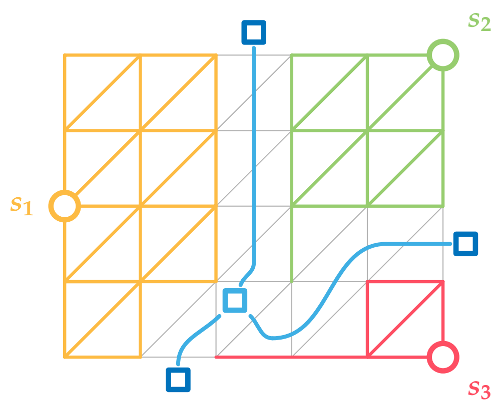


Figure X.1: A triangulated grid with 3 sites (s_1, s_2, s_3). The abstract dual Voronoi multigraph $\mathcal{M}$ (blue lines) partitions the graph into 3 regions, meeting at a central Voronoi vertex face (light blue square). For each site, the edges of its Voronoi cell are colored. The dark blue squares represent the same unique vertex corresponding to the outer face.

The attachment vertices. We define one attachment vertex for each Voronoi cell-Voronoi vertex adjacency. Let f be a Voronoi vertex of G. By definition, the boundary of f contains vertices belonging to at least three distinct Voronoi cells.

For each Voronoi cell $\mathrm{Vor}(s)$ that intersects the boundary of f, we arbitrarily select exactly one vertex $x \in \mathrm{Vor}(s)$ that lies on the boundary of f. We define x to be the *attachment vertex* of cell s at the Voronoi face f.

The set of attachment vertices X_s for a site s is the collection of all such chosen vertices in $\mathrm{Vor}(s)$ across all Voronoi vertex faces incident to $\mathrm{Vor}(s)$. Because an attachment vertex is chosen exactly once for each incidence between the cell $\mathrm{Vor}(s)$ and a Voronoi vertex, the number of attachment vertices in X_s is exactly the number of incidences between the face corresponding to s in $\mathcal{M}$ and the vertices of $\mathcal{M}$. Equivalently, $|X_s|$ is the degree of the face corresponding to s in $\mathcal{M}$.

The web and the subtrees. Let $\widehat{T}_s$ be any BFS tree rooted at s in $G[\mathrm{Vor}(s)]$. We define T_s to be the *minimal subtree* of $\widehat{T}_s$ that contains the root s and all attachment vertices in X_s. Said differently, we remove any vertex in $\mathrm{Vor}(s)$ that does not lie on the unique path in $\widehat{T}_s$ between s and some attachment vertex $x \in X_s$.

The union of these minimal subtrees, $W = \bigcup_{s \in S} T_s$, forms the *web* of the Voronoi partition. The vertices of $G - W$ are contained within the interior of the regions bounded by W.

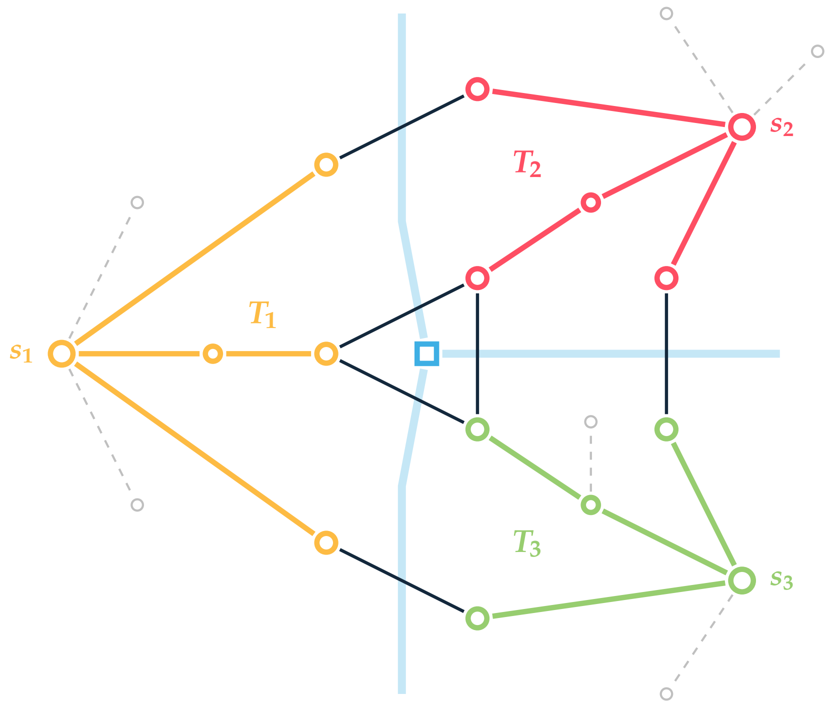


Figure X.2: Illustration of the web $W = \bigcup T_s$ in a planar graph. The dual multigraph $\mathcal{M}$ (thick blue) splits the graph, meeting at a central Voronoi vertex face (blue square) that borders cells $\mathrm{Vor}(s_1)$, $\mathrm{Vor}(s_2)$, and $\mathrm{Vor}(s_3)$. For each cell intersecting this face, exactly one attachment vertex is chosen on its boundary. The black edges represent the primal boundary edges. The thick colored edges represent the minimal subtrees T_s. Vertices not in W are shown in dashed gray.

We now prove that every web built from a distance-r dominating set S has small size with respect to r and $|S|$.

Lemma X.2.6. *Let $G = (V, E)$ be a connected planar graph, and let $S \subseteq V$ be a distance-r dominating set. Fix some Voronoi partition with respect to S (assuming some arbitrary strict total order on the sites), and let W be its web. Then $|V(W)| \leqslant (6r+1)|S| - 12r = O(r|S|)$.*

Proof. We first bound the total number of attachment vertices across all T_s for $s \in S$.

Recall that the vertices of $\mathcal{M}$ are the Voronoi vertex faces, its faces correspond to the sites S, and its edges represent the contiguous boundary segments separating adjacent Voronoi cells.

By construction, for each Voronoi vertex face f, and for each Voronoi cell $\mathrm{Vor}(s)$ intersecting f, exactly one attachment vertex $x \in \mathrm{Vor}(s)$ is chosen on the boundary of f. In the multigraph $\mathcal{M}$, this corresponds to a vertex-face incidence. The number of times the face s meets a vertex of

$\mathcal{M}$ is exactly the length of its facial boundary walk, which is its degree $d_{\mathcal{M}}(s)$. Thus, for every site s, the total number of attachment vertices in T_s is exactly $d_{\mathcal{M}}(s)$.

Because S is a distance-r dominating set, each attachment vertex is connected to its site s by a path of length at most r in the primal graph. As the minimal subtree T_s is contained in the union of these paths, $|V(T_s)| \leqslant r \cdot d_{\mathcal{M}}(s) + 1$. Summing over all sites $s \in S$, the total number of vertices in W is at most:

$$|V(W)| \leqslant \sum_{s \in S} (r \cdot d_{\mathcal{M}}(s)) + |S| = |S| + r \sum_{s \in S} d_{\mathcal{M}}(s) = 2r \cdot |E(\mathcal{M})| + |S|$$

by the handshaking lemma for faces.

Now, we apply Euler's formula to the planar multigraph $\mathcal{M}$ to bound $|E(\mathcal{M})|$. Because every Voronoi vertex is defined by the meeting of at least 3 Voronoi cells, every vertex in $\mathcal{M}$ has degree at least 3, i.e. $2|E(\mathcal{M})| \geqslant 3|V(\mathcal{M})|$. We claim that $|E(\mathcal{M})| \leqslant 3|S| - 6$. Indeed, if $F(\mathcal{M})$ denotes the faces of $\mathcal{M}$, Euler's formula for connected planar multigraphs gives $|V(\mathcal{M})| - |E(\mathcal{M})| + |F(\mathcal{M})| = 2$. Substituting the bound $|V(\mathcal{M})| \leqslant \frac{2}{3}|E(\mathcal{M})|$ into this formula and recalling that $|F(\mathcal{M})| = |S|$ gives: $\frac{2}{3}|E(\mathcal{M})| - |E(\mathcal{M})| + |S| \geqslant$ which implies $|E(\mathcal{M})| \leqslant 3|S| - 6$.

Substituting this into our bound for the web gives:

$$|V(W)| \leqslant 2r|E(\mathcal{M})| + |S| \leqslant 2r(3|S| - 6) + |S| = (6r + 1)|S| - 12r.$$

This proves that $|V(W)| = O(r|S|)$. □

X.2.3 *Taming planar graphs using Voronoi cells*

In this section, we present the structure theorem we use to successfully analyze the algorithm.

Theorem X.2.7. *Let $G = (V, E)$ be a connected planar graph, and let $S \subseteq V$ be a distance-r dominating set. Fix some Voronoi partition with respect to S (assuming some arbitrary strict total order on the sites), and let W be its web. Then there exists a partition $(C_1, \ldots, C_t)$ of the connected components C of $G - W$ such that:*

(a) The number of parts satisfies $t \leqslant 3|S| - 6$.

(b) For every $i \leqslant t$, the union of the connected components of C_i is separated from the rest of the graph by $O(r)$ vertices. More precisely, $|N_G[\bigcup C_i] \cap W| \leqslant 4r + 2$.

(c) Any connected component of $G - W$ has weak diameter in G bounded by $4r + 1$.

Proof. We begin by constructing the supergraph G^+ by augmenting G with dummy vertices and edges.

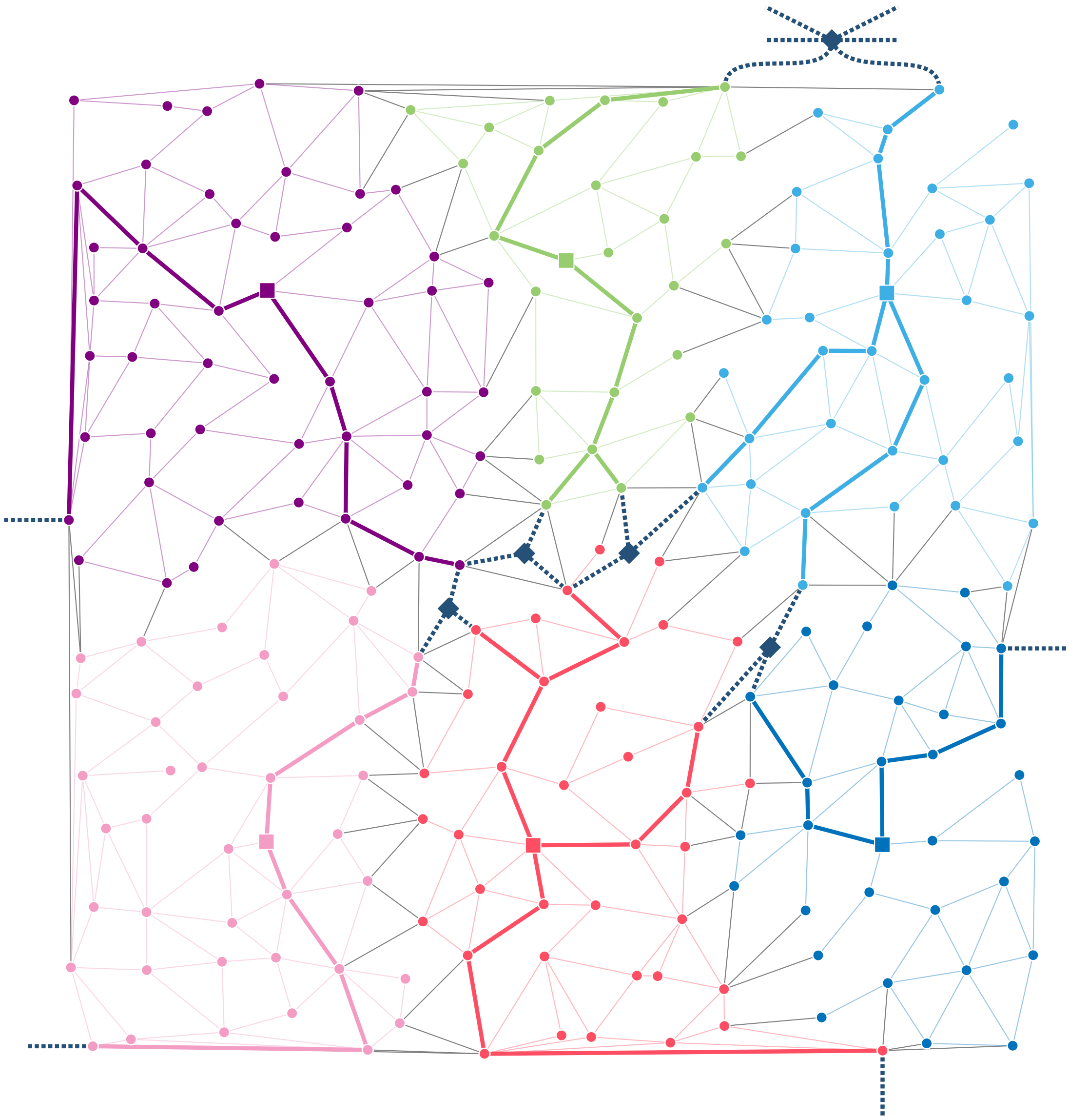


Figure X.3: A Voronoi partition and its extended web W^+. All vertices are colored according to their Voronoi cell. The thick colored edges represent the trees T_s, whose union forms the web W. Thin colored edges represent intra-cell edges not in W, while thin gray edges are inter-cell boundaries. The sites S appear as squares. Black diamonds represent the c_f's, placed inside faces where at least three Voronoi cells intersect. The dotted edges are the spokes, connecting all c_f to their respective attachment vertices. For visibility, spoke edges inside the outer face are broken into two dotted lines. The extended web W^+ consists precisely of the trees T_s and all spoke edges.

For every Voronoi vertex face f of G, we place a dummy center vertex c_f in the interior of the face. For each Voronoi cell $\mathrm{Vor}(s)$ that intersects f, we arbitrarily select exactly one attachment vertex $x \in \mathrm{Vor}(s)$ on the boundary of f. We add a *spoke* edge from c_f to every such attachment vertex. Because c_f connects only to the boundary of its own face, G^+ remains a planar graph. Let C be the set of all c_f's for every $f \in V(\mathcal{M})$.

Define the extended web W^+ as the plane subgraph of G^+ whose vertex set is $V(W) \cup C$ and whose edge set consists exactly of all edges of the minimal subtrees T_s together with all spoke edges. For an illustration, see Section X.2.3.

To understand the faces of W^+, we first look at its cycles. In W^+, the only edges connecting a tree T_s to vertices outside of T_s are the spokes connecting to the dummy vertices c_f. Because each T_s is a tree, any cycle in W^+ must strictly alternate between paths within the trees T_s and paths through the dummy vertices c_f. Specifically, a cycle must take the form of an alternating sequence: a path in T_{s_1}, a dummy vertex

c_1, a path in T_{s_2}, a dummy vertex c_2, etc...

Let k be the number of distinct trees T_s visited by such a cycle in W^+. Because each c_f connects to exactly one attachment vertex in any given tree T_s, a cycle cannot enter and leave a tree through the same dummy vertex. Thus, $k \geqslant 2$.

We claim that for any face f of W^+, the bounding cycle has exactly $k = 2$ (meaning it visits exactly two trees and two dummy vertices). Assume for sake of contradiction that a face of W^+ is bounded by a cycle with $k \geqslant 3$, which means it involves at least 3 distinct Voronoi cells $\mathrm{Vor}(s_1), \mathrm{Vor}(s_2), \mathrm{Vor}(s_3)$. The bounding cycle in W^+ forms a simple closed curve in the plane. Because W^+ is drawn as a subgraph of G^+, the interior of f corresponds to a region in the original plane graph G. By assigning to each vertex of G inside this region the "color" of its Voronoi cell, we can apply Sperner's Lemma (on a maximally triangulated supergraph of G). The boundary of the region alternates between at least 3 distinct Voronoi colors. By Sperner's Lemma, there must exist a face f^* of G within this region (which is bounded by f) whose boundary contains vertices of at least three distinct colors (Voronoi cells).

However, any face of G meeting three or more Voronoi cells is, by definition, a Voronoi vertex. By our construction of W^+, the interior of f^* contains a dummy vertex c_{f^*} connected by three distinct spoke edges to the attachment vertices $x_1 \in T_{s_1}$, $x_2 \in T_{s_2}$, and $x_3 \in T_{s_3}$, such that $x_i \in \mathrm{Vor}(s_i)$ for $i = 1, 2, 3$. In G, for each $i = 1, 2, 3$, the vertex x_i belongs to the shortest-path tree T_{s_i} rooted at s_i. Thus, there exists a path P_i in T_{s_i} from $V(f)$ to x_i. By planarity of G, these paths are entirely contained inside the face f, and separate it into three regions in W^+, which is impossible as f is a face in W^+.

Thus, every face of W^+ is bounded by a cycle with $k = 2$.

This means the boundary of any face in W^+ consists of:

- the spokes $x_2^a c_a$ and $c_a x_1^a$,
- a path P_1 in T_{s_1} between attachment vertices x_1^a and x_1^b,
- the spokes $x_1^b c_b$ and $c_b x_2^b$,
- a path P_2 in T_{s_2} between attachment vertices x_2^b and x_2^a.

Because S is a distance-r dominating set, the heights of T_{s_1} and T_{s_2} are at most r. The path P_1 connects two vertices in T_{s_1} via their lowest common ancestor, so its length is at most $2r$. The same holds for P_2.

Note that any set of vertices of G separated from the rest of the graph in G^+ by vertices in W^+ is also separated from the rest of the rest of the graph in G by vertices in W. This is because G^+ is a supergraph of G.

Now we bound the total number of faces of W^+. To do this, we first bound the number of spoke edges in W^+. To this end, notice that the total number of spoke edges in W^+ is exactly equal to the number of vertex-face adjacencies in $\mathcal{M}$.

In any planar graph, the number of vertex-face adjacencies is equal to the sum of the sizes (degrees) of all its faces. By the handshaking lemma for faces, this sum is exactly $2|E(\mathcal{M})|$.

Now, we apply Euler's formula to the planar multigraph $\mathcal{M}$ to bound $|E(\mathcal{M})|$. Because every Voronoi vertex face intersects at least 3 Voronoi cells, every vertex in $\mathcal{M}$ has degree at least 3, i.e. $2|E(\mathcal{M})| \geqslant 3|V(\mathcal{M})|$. We claim that $|E(\mathcal{M})| \leqslant 3|S| - 6$. Indeed, if $F(\mathcal{M})$ denotes the faces of $\mathcal{M}$, Euler's formula for connected planar multigraphs gives $|V(\mathcal{M})| - |E(\mathcal{M})| + |F(\mathcal{M})| = 2$. Substituting the bound $|V(\mathcal{M})| \leqslant \frac{2}{3}|E(\mathcal{M})|$ into this formula and recalling that $|F(\mathcal{M})| = |S|$ gives: $\frac{2}{3}|E(\mathcal{M})| - |E(\mathcal{M})| + |S| \geqslant 2$, which implies $|E(\mathcal{M})| \leqslant 3|S| - 6$.

Thus, W^+ contains at most $2|E(\mathcal{M})| \leqslant 6|S| - 12$ spoke edges.

As established earlier, every face of W^+ is bounded by a cycle with $k = 2$. This means the boundary walk of every face of W^+ strictly alternates between two tree paths and two dummy vertices c_f, and therefore contains exactly 4 spoke edges. Because W^+ is a plane graph and spokes are not bridge edges, every spoke edge lies on the boundary of exactly two faces of W^+. If N_S is the number of spoke edges, summing over all faces of W^+ gives $4|F(W^+)| = 2 \cdot N_s$. Thus, it follows that W^+ has at most $|F(W^+)| \leqslant 3|S| - 6$ faces.

Let $t = |F(W^+)|$ and construct the partition $(C_1, \ldots, C_t)$ where each C_i corresponds to a face of W^+ as follows. Let C be any connected component of $G - W$. It is contained in one of the faces of W^+ corresponding to C_i; put C in C_i. The fact that $t = |F(W^+)| \leqslant 3|S| - 6$ proves Item (a).

For any part C_i, all edges connecting C_i to W must have an end on a face of W^+. As showed previously, the size of a face in W^+ is at most $4r + 2$. Thus, the size of the separator $N_G[\bigcup C_i] \cap W$ is bounded by $4r + 2$, proving Item (b).

Finally, we bound the weak diameter of components in $G - W$. Consider some connected component $C \in C_i$. C_i corresponds to a face of W^+, and thus intersects only two Voronoi cells, $\mathrm{Vor}(s_1)$ and $\mathrm{Vor}(s_2)$, which are at distance at most $2r$ (otherwise another site would be contained in this face, which is impossible). Let $x, y \in C$. Both vertices are assigned respectively to the sites $s_x, s_y \in \{s_1, s_2\}$, with $d_G(x, s_x), d_G(y, s_y) \leqslant r$. As said previously, we have $d_G(s_x, s_y) \leqslant 2r + 1$, and by the triangle inequality, $d_G(x, y) \leqslant 4r + 1$. Thus, the weak diameter of C is $O(r)$, proving Item (c). □

X.2.4 Proof of Theorem X.2.5

Proof. Let G be a labeled planar graph, and let r be the domination distance. Let Y be a global distance-r minimum dominating set of G that is "best" according to the lexicographical ordering defined before the algorithm. Let D be the distance-r dominating set of G returned by Algorithm 8.

For every vertex $u \in V(G)$, let D_u be the dominating set computed by u. By definition of the algorithm:

- D_u contains no redundant vertices (it is "nice" in the sense that no vertex's r-neighborhood is strictly contained in another's).

- D_u is the lexicographically smallest optimal distance-r dominating set of $N^{4r+1}[u]$.
- v_u is the vertex of smallest label in $D_u \cap N^r[u]$, and $D = \bigcup_{u \in V(G)} \{v_u\}$.

Since every vertex u adds a vertex $v_u \in N^r[u]$ to D, the set D is trivially a valid distance-r dominating set for G. It remains to bound the size of D as a function of $|Y|$.

By Theorem X.2.7 (applied to the global optimum Y), the connected components of $G - W$ (where W is the web of Y) can be partitioned into $t \leqslant 3|Y| - 6$ distinct parts $C_1, \ldots, C_t$. For each part C_i, its boundary $B_i = N_G[\bigcup C_i] \cap W$ has size at most $4r + 2$, and the weak diameter of $C_i \cup B_i$ in G is bounded by $4r + 1$.

The key observation relies on the local consistency of the lexicographically smallest sets. Let C_i be one of these part and $C \in C_i$ be a connected component of $G - W$. As the weak diameter of C is at most $4r + 1$, the set $N_G[C] \cap W$ which separates C from the rest of the graph is fully contained within every closed neighborhood $N_G^{4r+1}[u]$ for any $u \in C$. Specifically, the boundary B_i completely isolates C_i from the rest of the graph.

When any $u \in C_i$ computes its local optimal set D_u for $N^{4r+1}[u]$, the choice of v_u depends exclusively on the amount of domination coverage provided by vertices on the boundary. More precisely, the value of $D_u \cap (\bigcup C_i \cup B_i)$ is completely determined by the distances of each boundary vertex $w \in B_i$ to the closest chosen dominating vertex in D_u. Said differently, there is a unique best way to extend a distance-r dominating set of $G - \bigcup C_i$ to $\bigcup C_i \cup B_i$, once those distances are specified. Because we are considering distance-r domination, this can be summarized by an integer in $\{1, \ldots, r+1\}$ (where $r + 1$ indicates that the boundary vertex is not dominated). Therefore, there are at most $(r + 1)$ relevant states for each vertex in B_i. Fixing this state vector for B_i completely determines the unique, lexicographically smallest way to extend this partial solution to optimally dominate C_i.

Since B_i contains at most $4r + 2$ vertices, there are at most $(r + 1)^{4r+2}$ possible state vectors for the boundary that any local solution could induce. Consequently, across all vertices $u \in \bigcup C_i$, the set $D_u \cap (\bigcup C_i \cup B_i)$ can take at most $(r + 1)^{4r+2}$ distinct values. For each specific boundary state vector, the size of $D_u \cap C_i$ is always at most 2.

Now, let us bound the number of distinct vertices contributed to D by the algorithm. A vertex u adds v_u to D. The vertex v_u can either belong to the web W or to some part C_i. By Lemma X.2.6, there are $O(r|Y|)$ vertices in W. Now, for a fixed part C_i, the number of distinct vertices added to $D \cap \bigcup C_i$ by vertices $u \in \bigcup C_i$ depends on the number of distinct local solutions $D_u \cap (\bigcup C_i \cup B_i)$. As established, there are at most $(r + 1)^{4r+2} = r^{O(r)}$ such distinct configurations. Therefore, the total number of distinct vertices contributed to $D \cap (\bigcup C_i \cup B_i)$ across all $u \in C_i$ is bounded by $r^{O(r)} \cdot |D_u \cap (\bigcup C_i \cup B_i)| = r^{O(r)}$ as we always have $|D_u \cap (\bigcup C_i \cup B_i)| \leqslant 2$.

Summing over all $t \leqslant 3|Y| - 6$ parts, the total number of vertices contributed to D by vertices inside the parts C_i is bounded by:

$$\sum_{i=1}^{t} r^{O(r)} \leqslant (3|Y| - 6) \cdot r^{O(r)} = r^{O(r)}|Y|.$$

Moreover, the total number of vertices contributed to D by vertices inside W is bounded $|W| = (r|Y|)$.

Putting everything together, we get that $|D| = O\left(r^{O(r)} \cdot |Y|\right)$, proving the approximation ratio. Because the algorithm only requires each vertex to explore its $(5r+1)$-neighborhood, this shows that Algorithm 8 is a $k^{O(k)}$-approximation algorithm that runs in $O(r)$ rounds. □

X.3 MDS on planar graphs

Let us give a brief overview of the literature for the Minimum Dominating Set problem for planar graphs in the LOCAL model. Lenzen, Pignolet, and Wattenhofer [LPW13] proposed the first constant-factor approximation algorithm. The analysis of their algorithm was later improved by Wawrzyniak [Waw14] to achieve an approximation ratio of 52. More recently, Heydt et al. [HKO+25] introduced a new algorithm that further improves this ratio to $11 + \varepsilon$, for any $\varepsilon > 0$. [2]

[2] With a running time that is a function of ε only.

On the hardness side, the current best lower bound is $7 - \varepsilon$ [HLS14]. It is unlikely that this specific technique can give a lower bound greater than 7 for planar graphs. To establish their bound, Hilke, Lenzen, and Suomela use an algorithm-dependent construction: for any given algorithm that outputs a dominating set, the authors construct a specific graph on which the algorithm is forced to select almost all vertices. In these constructions, some minimum dominating set D intersects the closed neighborhood of almost every vertex exactly once, i.e. $|N[v] \cap D| = 1$ for every $v \in V(G)$. However, the set D' returned by the algorithm contains the entire closed neighborhood, such that $|N[v] \cap D'| = |N[v]| = \deg(v) + 1$ for all but an ε-fraction of the vertices. Therefore, the resulting lower bound on the approximation ratio is roughly equal to the average degree of the graph plus one. Because the average degree of planar graphs is at most 6, one cannot hope for a lower bound higher than 7.

For these reasons, we conjecture that the right answer is 7. Note that this is a particular case of the bolder Conjecture V.1.3.

Conjecture X.3.1. *There is a constant-round 7-approximate LOCAL algorithm for Minimum Dominating Set on planar graphs.*

References

[ÅFP+09] Matti Åstrand et al. "A local 2-approximation algorithm for the vertex cover problem". In: *International symposium on distributed computing*. Springer. 2009, pp. 191–205.

[Ang80] Dana Angluin. "Local and global properties in networks of processors". In: *Proceedings of the twelfth annual ACM symposium on Theory of computing*. 1980, pp. 82–93.

[AW21] Saeed Akhoondian Amiri and Ben Wiederhake. "Distributed distance-r covering problems on sparse high-girth graphs". In: *International Conference on Algorithms and Complexity*. Springer. 2021, pp. 37–60.

[GKM+21] Paweł Gawrychowski et al. "Voronoi Diagrams on Planar Graphs, and Computing the Diameter in Deterministic $O(n^{5/3})$ Time". In: *SIAM Journal on Computing* 50.2 (2021), pp. 509–554.

[HKO+25] Ozan Heydt et al. "Distributed domination on sparse graph classes". In: *European Journal of Combinatorics* 123 (Jan. 2025), p. 103773. DOI: 10.1016/j.ejc.2023.103773.

[HLS14] Miikka Hilke, Christoph Lenzen, and Jukka Suomela. "Brief announcement: local approximability of minimum dominating set on planar graphs". In: *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*. PODC '14. Paris, France: Association for Computing Machinery, 2014, pp. 344–346. ISBN: 9781450329446. DOI: 10.1145/2611462.2611504. URL: https://doi.org/10.1145/2611462.2611504.

[LPW13] Christoph Lenzen, Yvonne-Anne Pignolet, and Roger Wattenhofer. "Distributed minimum dominating set approximations in restricted families of graphs". In: *Distributed Computing* 26.2 (Mar. 2013), pp. 119–137. DOI: 10.1007/s00446-013-0186-z.

[LW24] Christoph Lenzen and Sophie Wenning. *Tight Bounds for Constant-Round Domination on Graphs of High Girth and Low Expansion*. Tech. rep. 2408.12998v1 [cs.DC]. arXiv, Aug. 2024. DOI: 2408.12998v1.

[Waw14] Wojciech Wawrzyniak. "A strengthened analysis of a local algorithm for the minimum dominating set problem in planar graphs". In: *Information Processing Letters* 114.3 (Mar. 2014), pp. 94–98. DOI: 10.1016/j.ipl.2013.11.008.

Chapter XI
Conclusion

Part A: decision problems on graphs of high degree. In Part A, we took a first step toward understanding the distributed complexity of graph problems as a function of both the number of nodes and the degrees of the graph. We introduced the class of structurally simple binary labeling problems, proved a complete classification of this class in the logarithmic regime, showed that all families defined by context-free languages are structurally simple, and developed a more aggressive version of the rake-and-compress technique that yields upper bounds benefiting from high-degree nodes.

Several questions remain open. In particular, while our classification is constructive, it is still unclear how difficult it is to decide, from context-free grammars defining a problem family, whether the complexity is $O(\log_d n)$ for all but finitely many values of d and δ.

> **Question II.1.6.** *Given context-free grammars for thin languages $\hat{W}$ and $\hat{B}$ that define a problem family $\Pi(d, \delta)$, how hard is it to decide if the complexity of $\Pi(d, \delta)$ is $\Theta(\log_d n)$ for all but finitely many values of d and δ?*

More broadly, our results suggest that structurally simple families capture a large class of natural problems. We conjecture that the aggressive rake-and-compress framework developed here extends beyond binary labeling problems and will prove useful for the study of more general local problems in high-degree graphs.

Finally, we leave open the hard question of classifying the complexities of LCL problems for unbounded degree graphs. Note that recently, a paper of Schmid [Sch26] answers this question for the polynomial regime, i.e. complexities of the form $\Theta(n^{1/k})$.

Part C: a structural interlude. In Chapter VII, we proved that every graph whose edges can be covered by k shortest paths has pathwidth $O(k^4)$, improving the previous exponential upper bound and answering a question of Dumas, Foucaud, Perez and Todinca. We also showed a tight bound of k for $k \leqslant 3$. Finally, we established that although vertex-coverability by two isometric trees does not force bounded treewidth, every graph whose edges are covered by two isometric trees has treewidth at most 2. Several questions remain open. The main one is whether the polynomial upper bound obtained here can be improved to a linear one, and more specifically whether every graph edge-coverable by k shortest paths has pathwidth $O(k)$, or even pathwidth at most k. Another natural open problem is whether graphs vertex-coverable by k shortest paths also admit a polynomial bound on their pathwidth, since the techniques developed here do not seem to extend to that setting. On the side of isometric trees, it is known that no analogue can hold in full generality for $k \geqslant 4$, but the case of graphs edge-coverable by 3 isometric subtrees and having large treewidth remains open.

Parts B and C: approximating in the LOCAL *model.* In Chapter III, we developed a broader perspective on constant-round distributed approximation beyond planar graphs. On the conceptual side, we introduced a general framework based on asymptotic dimension, local niceness, and uniformity, and established meta-theorems (Meta-Theorem I, Meta-Theorem II and Meta-Theorem III) that transfer local approximation algorithms from a well-understood graph class to larger sparse classes. On the algorithmic side, this led to new constant-ratio approximation algorithms for minimum dominating set and related problems on bounded-Euler-genus and more general locally nice graph classes (see Corollaries III.4.3 and III.4.5), while isolating cutability as a key structural property behind these transfers (see Proposition III.7.7).

This work can lead to multiple research directions. There is a related question I would like to highlight. We have developed meta-theorems for algorithms that use polynomial identifiers to approximate optimization problems. However, as of this moment, we are unaware of such constant-time algorithms truly needing polynomial identifiers. Therefore, an interesting question is: can some problems force this? Can any such algorithm be transformed into an algorithm that is agnostic to polynomial identifiers?

We then investigated how asymptotic dimension can be used in the context of minor-closed classes. Our lower bound of Chapter V (see Theorem V.1.1) shows that for minimum dominating set on H-minor-free graphs, one cannot hope for a constant-round approximation ratio smaller than $2\,\mathrm{pw}(H) - 1$. In particular, this indicates that the pathwidth of the excluded minor is the relevant parameter governing the constant-round approximability of the problem, rather than simply the number of vertices of H.

This viewpoint is further supported by positive results. In Chapter VIII, building on the techniques of the $K_{2,t}$-minor-free case of Chapter IV (see Theorem IV.3.1), we proved in Theorem VIII.1.1 that the conjectured behavior holds when $\mathrm{pw}(H) = 2$: every graph class excluding a minor of pathwidth at most two admits a constant-round constant-ratio approximation algorithm for minimum dominating set. Taken together, these results suggest that pathwidth is the true complexity parameter for domination in minor-closed graph classes.

Conjecture V.1.2. *For every graph H, there is an $O(\mathrm{pw}(H))$-approximation $O_H(1)$-round* LOCAL *algorithm for* Minimum Dominating Set *on H-minor-free graphs.*

The main open problem is now to close the gap between the general lower bound and the known upper bounds for arbitrary excluded minors. If Conjecture V.1.2 is true, then for the particular case $H = K_{s,t}$ with $s < t$, there exists an $f(t)$-round $O(s)$-approximation LOCAL algorithm for minimum dominating set on $K_{s,t}$-minor-free graphs, which would be optimal up to constant factors.

Beyond this conjecture, an important direction is to determine whether the machinery developed here can be extended to wider classes of optimization problems. In particular, for every additive cuttable minimization problem Π, does there exist a local notion of *locally critical* vertex with the following property? For every graph class $\mathscr{C}$ of bounded asymptotic dimension, there is a constant $c_{\Pi,\mathscr{G}}$ such that every graph $G \in \mathcal{G}$ contains at most $c_{\Pi,\mathcal{G}} \cdot \mathrm{OPT}_\Pi(G)$ locally critical vertices. Moreover, these vertices capture all the global difficulty of the instance, in the sense that once they are selected, removed, or otherwise handled, the remaining graph decomposes into connected components of bounded weak diameter. In particular, one may then hope to solve in a local way the leftover instance independently inside each component. For Minimum Dominating Set, this notion of local criticality is realized by (interesting) vertices contained in local cuts.

Part D: perspectives, and other problems. Returning to Conjecture V.1.2, we explored in Chapter IX several ideas that may lead to a proof of the conjecture. We lay out a proof plan for both technical challenges: first, bounding the number of interesting vertices in l-cuts (for $k \geqslant 3$), and secondly, proving that the leftover vertices, i.e. the vertices undominated by those interesting vertices, form connected components of bounded weak diameter.

In Chapter X, we tackle problems beyond Minimum Dominating Set, such as Minimum Vertex Cover in degenerate graphs and Minimum Distance-k Dominating Set in bounded genus graphs. We showed that degenerate graphs admit a constant time $(2+\varepsilon)$-approximation algorithm (see Theorem X.1.3).

We also showed in Theorem X.2.5 that Minimum Distance-k Dominating Set admits an $O(k)$-round $k^{O(k)}$-approximation for planar graphs of arbitrary girth. This approximation ratio is probably not tight, and we leave open the question of reducing it. We further conjecture an analogous statement of the MDS conjecture for Minimum Distance-k Dominating Set.

Conjecture X.2.4. *There exists a function f such that for every graph H, there is an $O(f(k) \cdot \mathrm{pw}(H))$-approximation $O_H(1)$-round* LOCAL *algorithm for* Minimum Distance-k Dominating Set *on H-minor-free graphs.*

Finally, the optimal approximation ratio for Minimum Dominating Set on planar graphs is still unknown. We conjecture that the right answer is 7.

Conjecture X.3.1. *There is a constant-round 7-approximate* LOCAL *algorithm for* Minimum Dominating Set *on planar graphs.*

References

[Sch26] Gustav Schmid. "LCLs Beyond Bounded Degrees". In: *arXiv preprint arXiv:2602.02340* (2026).