université de BORDEAUX

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE

DE MATHEMATIQUES ET INFORMATIQUE

SPÉCIALITÉ : INFORMATIQUE

Par **Paul BASTIDE**

Structures locales en combinatoire : reconstruction, saturation, dispersion et universalité.

Sous la direction de : **Marthe BONAMY**
Co-directrice : **Carla Groenland**

Soutenue le 26 Juin 2025

Membres du jury :

Mme. Marthe Bonamy	Chargée de Recherche	Université de Bordeaux	Directrice
Mme. Mireille Bousquet-Mélou	Directrice de Recherche	Université de Bordeaux	Présidente
Mme. Carla Groenland	Professeure assistante	Technische Universiteit Delft	Co-directrice
M. Gwenaël Joret	Professeur	Université Libre de Bruxelles	Rapporteur
M. Imre Leader	Professeur	University of Cambridge	Rapporteur
M. Frédéric Magniez	Directeur de Recherche	IRIF	Examinateur
M. Pawel Pralat	Professeur	Toronto Metropolitan University	Rapporteur
Mme. Dana Ron	Professeure	Tel Aviv University	Examinatrice

Structures locales en combinatoire: reconstruction, saturation, dispersion et universalité.

Résumé : Cette thèse explore les structures locales en combinatoire. Elle se concentre sur l'influence de sous-structures locales sur les propriétés globales des objets étudiés. Elle est divisée en quatre parties, chacune examinant un problème different sous le prisme de leur sous-structures locales: la reconstruction via oracle de distance, la saturation de posets, la dispersion pour les plongements, et les structures universelles.

Dans le chapitre consacré à la reconstruction par oracle de distance, on cherche à reconstruire un graphe inconnu à l'aide d'informations locales. On suppose uniquement l'accès à un oracle fournissant la distance d'un plus court chemin entre deux sommets choisis. Le problème est étudié pour différentes classes de graphes et nous améliorons à la fois les meilleures bornes inférieures et/ou supérieures connues. En particulier, dans ce manuscrit nous proposons un algorithme optimal pour la reconstruction des arbres et un algorithme quasi-optimal pour la reconstruction des graphes sans longs cycles induits.

Dans un second temps, nous étudions la saturation de posets, introduite par Katona et Tarjan en 1981. Ce concept généralise les travaux de Turán en théorie extrémale des graphes. Une famille est P-saturée si elle est maximale et qu'elle évite toute sous-structure localement isomorphe à P (c'est-à-dire qu'elle évite P comme sous-poset induit). Nous nous intéressons au "nombre de saturation" d'un poset P, défini comme la taille minimale d'une famille P-saturée incluse dans l'hypercube de dimension n. Malgré des avancées significatives sur ce problème ces dernières années, le nombre de saturation n'est connu exactement que pour de petits posets, et il reste largement incompris. Dans cette thèse, nous explicitons, grâce à une généralisation d'un lemme de Lehman et Ron, la valeure exacte du nombre de saturation de l'antichaîne de taille k. Nous établissons également la première borne supérieure générale sur les valeurs possible du nombre de saturation pour tout poset P.

Dans le chapitre sur les "plongements dispersés", nous étudions la répartition des arbres couvrants dans les graphes denses. À première vue, ce sujet peut sembler éloigné de l'axe globale de la thèse, les structures locales, puisqu'une structure "couvrante" est par définition globale. Cependant, la technique clé que nous développons dans ce manuscrit pour construire ces structures couvrantes repose sur une subdivision en plusieurs sous-structures locales. Grâce à cette nouvelle méthode, nous obtenons une démonstration plus simple et plus flexible de l'existence de "plongements dispersés" pour les arbres de degré borné. En particulier, notre approche ne fait pas appel au lemme de régularité de Szemerédi.

Enfin, nous étudions les structures universelles. Étant donné une famille $\mathcal{F}$ d'objets combinatoires et un entier n, un autre objet U est dit universel s'il contient chaque élément de $\mathcal{F}$ de taille n comme sous-structure locale. On appelle une telle structure "fidèle" lorsque U appartient également à F. Dans ce chapitre, on s'intéresse à la taille minimale des graphes universels "fidèles" pour les classes de graphes closes par mineurs ainsi qu'à la taille de posets universels pour la famille contenant tous les posets.

Mots-clés : Reconstruction, Saturation, Structures locales, Dispersion, Universel

---

LaBRI
UMR 5800 Université, 33000 Bordeaux, France.

Local structure in combinatorics: reconstruction, saturation, spreadness and universality

Abstract: This thesis explores local structures in combinatorics, focusing on understanding how local configurations influence the global properties and structures of combinatorial objects. This thesis is divided into four parts, each investigating different problems related to local structures: distance reconstruction, poset saturation, spread embedding, and universal structures.

In the chapter related to distance reconstruction, we seek to reconstruct an unknown graph using local information. We only have access to an oracle providing the shortest path distances between pairs of vertices. We study this problem for different classes of graphs, improving both the best-known lower bounds and upper bounds in various cases. In particular, we provide an optimal algorithm for reconstructing trees and a near-optimal algorithm for reconstructing graphs without long induced cycles.

Then, we study poset saturation, introduced by Katona and Tarjan in 1981. This concept extends Turán's seminal work in extremal graph theory. A P-saturated family is a maximal family that avoids any substructure locally isomorphic to P (i.e., avoids P as an induced subposet). We study the saturation number of a poset P, defined as the minimum size of a P-saturated family included in the hypercube of dimension n. Despite substantial work on this problem in recent years, the saturation number is only known exactly for small posets, and its possible behaviors are still largely unknown. In this thesis, we will show that an extension of a lemma from Lehman and Ron allows us to compute exactly the saturation number of the antichain of size k. We also prove the first general upper bound on the possible behavior of the saturation number for any poset P.

In the chapter on spreadness, we study the distribution of spanning trees in dense graphs. It might seem, at first, far from local, as a "spanning" structure is by definition global, but the key technique we developed to construct these global spanning structures goes through the subdivision into multiple local substructures. Via this novel method, we are able to give a simpler and more flexible proof of the existence of spread embeddings for bounded-degree trees. In particular, our approach does not use Szemerédi's Regularity Lemma.

Finally, we study universal structures. Given a family $\mathcal{F}$ of combinatorial objects and an integer n, another object U is said to be universal if it contains every element of $\mathcal{F}$ of size n as a local substructure. We call such a structure "faithful" when U is also part of $\mathcal{F}$. In this chapter, we study the minimum size of faithful universal graphs for minor-closed classes of graphs as well as the size of universal posets for the family of all posets.

Keywords: Reconstruction, Saturation, Structures locales, Dispersion, Universel

---

LaBRI
UMR 5800 Université, 33000 Bordeaux, France.

Acknowledgements

J'ai toujours détesté les analyses de cas trop longues. Si je ne veux pas faire de ces remerciements une liste de noms, c'est en partie pour cela, mais surtout car j'aimerais remercier tous les gens qui se sentent concernés par les quelques mots ci-dessous. Si je n'écris aucun nom, j'espère—peut-être naïvement—que personne ne se sentira oublié·e. Donc, pour tous ceux qui lisent ces mots, merci.

Firstly, I would like to sincerely thank the members of my jury for having taken the time to read my thesis and for their thoughtful questions, comments, and advice.

I would like to thank my advisors. Even though I know both of you are fluent French speakers, I still feel somewhat uncomfortable speaking in French with one of you, so I will continue in English. Even in French, it would be hard to express just how grateful I am and how lucky I feel to have had you both as advisors. The literal meaning of encadrer in French is "to frame", which poorly captures the role you've both played over the past three years. You have always given me all the freedom I could have wished for, while being there whenever I needed help and advice, always selflessly looking out for what was best for me. Thank you for your guidance, your patience, and all your time.

J'aimerais aussi remercier les deux membres de mon comité de suivi de thèse ainsi que mon « troisième » encadrant, qui m'ont suivi du début jusqu'à la fin de cette thèse et ont souvent pris de leur temps pour me donner leurs conseils.

Thanks to every one of my co-authors. I met so many incredible people through research—people I laughed with, people I climbed with, people I played "le jeu des robots" with, people I hiked with. Surprisingly, sharing a two-person tent with two other people for five days creates a strong bond. Thank you all for those moments.

Merci à mes grands frères et grandes sœurs de thèse, pour vos conseils et votre bienveillance. Merci à mon mentor d'escalade, c'est encore souvent vers toi que je me tourne lorsque j'ai besoin d'un conseil. Merci à mes petits frères, parce qu'ils sont géniaux, tout simplement. Merci aussi d'avoir trouvé des références en allemand pour résoudre mes problèmes d'arithmétique.

During those three years, I had two homes: one in Bureau 267 and one in HB.04.250. I moved back and forth quite a lot, and living this rootless life was sometimes challenging. But each time I returned, you made it feel like I had never left.

Aux habitants du Bureau 267, merci. J'en connais une depuis six ans, bien avant le début de notre thèse. J'aimerais te remercier, pour avoir été là quand j'avais besoin de discuter, pour ta gentillesse et ta franchise. Le grand Bordelais cinéphile, sache que tu es une personne rare. Le bureau n'aurait définitivement pas été pareil si tu n'avais pas été là, tous les matins lorsque j'ouvrais la porte. Je pense que le poster de Zizou est officiellement la meilleure blague de ma thèse. Merci aussi au doyen du bureau, qui balançait ses cartes One Piece

aussi vite que ses blagues. Merci de nous avoir acceptés et accueillis dans ton antre en première année. Et enfin, merci à la dernière arrivée, avec son sourire et sa bonne humeur contagieuse !

To the inhabitants of HB.04.250, thank you. As multicultural as this office might be, it always felt like home. Thank you for all the dinners we shared and every sports evening. Thanks to our office, I gained a climbing partner, two Chinese teachers, a fellow banana bread enjoyer, and even a bike savior. Let's make sure to share another dinner next time I'm around Delft!

Merci à tous mes amis, pour votre justesse, votre intelligence, votre gentillesse, et tous vos conseils. J'ai beaucoup grandi pendant ces trois ans, et c'est majoritairement grâce à vous. Merci en particulier, aux ames charitables, qui m'ont accompagné grimper, en voitures comme à vélo.

Et enfin, merci à ma famille, à mes parents qui m'ont toujours soutenu, à mes grands-parents qui m'ont toujours nourri, et à ma sœur qui m'a toujours secoué aux moments où j'en avais le plus besoin. Tu es la meilleure docteure de notre famille — et je suis fier de toi.

To the calm in the storm, and the wind that kept me moving.

Contents

Introduction

This chapter provides a broad introduction to the topics covered in this thesis, along with brief motivations for their study. It is available both in English and in French.

Extended introduction in english

A *graph* G consists of a set $V(G)$ of abstract objects called *vertices* and a set $E(G)$ of one-to-one "connections" between those vertices denoted by pairs of elements in $V(G)$. This simple mathematical structure can help describe a wide range of real-life situations. For example, we can consider the graph G associated with a social network where the vertices represent the registered people and the edges represent the "friend" relationship. Once we have constructed this mathematical object, we can translate concrete questions about the social network into graph-theoretic questions about G. Suppose you want to find out who form your largest "friend group". This would correspond to identifying the largest set of vertices that are all connected by edges and include the vertex representing you.

But the versatility of graphs extends far beyond network representation. The LaTeX compiler I'm using right now to write this manuscript relies on graphs to allocate variables to registers. The route I'll take home later, after a productive day of writing, was designed using graphs. And when I finally get home and decide to unwind in the evening, browsing through movie options on my favourite streaming platform—that, too, will be optimised using graphs.

Graph theory is one of the oldest branches of *Combinatorics*, itself a branch of mathematics. While no official definition of Combinatorics has been universally accepted, we will describe it here as the study of discrete structures through enumeration, construction, optimisation, and analysis. This manuscript studies various problems in Combinatorics often related to graph theory. We will now give a broad introduction to the four terms forming the title of this thesis: Reconstruction, Saturation, Spreadness and Universality.

Reconstruction Have you ever wondered what the best questions to ask in the board game *Guess Who?* are? For those unfortunate few who missed out on this childhood classic, *Guess Who?* is a two-player game in which players take turns asking yes/no questions to deduce the identity of the character chosen by their opponent. This game serves as a perfect example of a reconstruction problem in combinatorics. We begin with a large set of characters, one of which has been secretly and perhaps even maliciously selected by our opponent. Our goal is to determine, or in mathematical terms, reconstruct, which character was chosen. To achieve this, we can only ask restricted yes/no questions – *queries* – that provide partial information about the character's identity. The challenge is that our opponent is trying to do the same, and we are effectively racing against them. Each question comes at a cost, and the objective is to deduce the answer faster than our opponent, that is, using as few questions as possible.

If you are a mathematician – or just someone who enjoys ruining fun with logic – you might come up with an arguably optimal strategy for *Guess Who?*, ensuring that, if the game starts with n possible characters, you can always guess your opponent's character in at most $\lceil \log_2 n \rceil$ questions. Moreover, this is optimal; if you ask strictly fewer questions, there will always exist two characters that you cannot distinguish.

Mathematicians are working to prove analogous results for more complex reconstruction problems. For example, one topic of this manuscript is the study of how many queries are required to reconstruct phylogenetic trees, using DNA-sequence distance comparisons instead of yes/no questions. A better understanding of this process could lead to significant savings of both time and cost in phylogenetic research.

Reconstruction questions generally fall into two categories: *existence* and *optimisation*. *Existence* questions ask whether an object can be uniquely reconstructed, even with an unlimited number of queries. If reconstruction is possible, the natural follow-up is an *optimisation* question: what is the minimum number of queries needed to determine the object uniquely?

One of the oldest and most famous graph reconstruction problems is an *existence* question, often referred to as the *Graph Reconstruction Conjecture* formulated by Stanisław Ulam and Paul Kelly independently in the late 1950s [113, 179]. It asks whether a simple, finite graph G with at least three vertices can be uniquely determined (up to isomorphism) by its "deck", defined as the multiset of subgraphs of the form $G \setminus v$ for every vertex $v \in V(G)$. Despite substantial effort from the research community and positive answers for restricted graph classes—see the following survey by Harary [88]—the problem is still open for general graphs.

Another long-standing problem, in the *optimisation* category, is the *evasiveness conjecture* also known as the Aanderaa–Karp–Rosenberg conjecture, named after the three researchers who introduced the conjecture around 1973 [163]. This problem extends well

beyond graph theory, but we will focus on its original graph-theoretic formulation here. The hidden object is the edge set of a graph G, and the questions we are allowed to ask, called *edge queries*, are of the form: Given $u, v \in V(G)$, "Is there an edge between u and v in G?". A graph class $\mathcal{G}$ is called *monotone* if, for any $G \in \mathcal{G}$, every graph H obtained from G by removing edges also belongs to $\mathcal{G}$. The conjecture states that for any nontrivial[1] monotone n-vertex graph class $\mathcal{G}$, determining whether a given G belongs to $\mathcal{G}$ requires $\binom{n}{2}$ edge queries in the worst case. Informally, this means that any algorithm must check *all* possible pairs of vertices before deciding whether $G \in \mathcal{G}$. The conjecture remains open and has only been proven for n a prime power by Kahn, Saks and Sturtevant [104] in 1984.

Saturation Suppose that you are tasked with designing a computer network for a company with the following two requirements. The first is to use as few communication links as possible, and the second is that specific computers must not be able to communicate directly for security reasons. If we represent the computers as vertices of a graph and the communication links as edges, the company's request translates into solving a saturation problem. More generally, a saturation problem involves finding a combinatorial structure that maximises or minimises a global cost function—in our example, the number of links—while satisfying local constraints, in our example, the forbidden links.

Many fundamental problems in combinatorics have been, or can be, framed as saturation problems. For example, in structural graph theory, *critical* graphs, introduced around 1950 by G. A. Dirac, are saturated structures that have been extensively studied[2] to understand the behaviour of the chromatic number. In extremal combinatorics, multiple foundational results emerge from the study of H-saturated graphs, and date back more than a century ago, starting with Mantel's works on triangle-free graphs [135] and continuing with the highly celebrated Turán's theorem [178] and Erdős, Hajnal, and Moon's theorem [68]. An H-saturated graph is a graph G such that H is not a subgraph of G but adding any edge to G creates a copy of H in G. Thanks to the two theorems cited above and an extensive effort of the research community — see the dynamic survey of Faudree, Faudree and Schmitt [71] — we now possess a deep comprehension of the edge counts in H-saturated graphs.

Recently, researchers have begun investigating variants and generalisations of H-saturated graphs. In particular, there has been growing interest in the behaviour of *H-induced-saturated* graphs, where the constraint of containing H as a subgraph is replaced by the requirement that H appears as an induced subgraph. This modification significantly increases the complexity of the problem, and for most graphs, the very existence of such saturated structures remains unknown [137, 32, 16]. In this thesis, we study saturation in partially ordered sets. Given a poset P, we investigate the minimal size of a P-saturated

[1] We exclude the empty class and the class containing all graphs.

[2] We refer the reader to the recent book by Stiebitz, Schweser, and Toft [173] for a survey on the subject.

family $\mathcal{F} \subseteq 2^{[n]}$, defined as a family $\mathcal{F}$ that does not contain P as an induced subposet, but for every $X \in 2^{[n]} \setminus \mathcal{F}$, the family $\mathcal{F} \cup \{X\}$ does. This line of research traces back to a 1981 result of Katona and Tarjan [112] and was later formalised by Gerbner, Keszegh, Lemons, Palmer, P'alv"olgyi, and Patk'os [82] for the non-induced version and by Ferrara, Kay, Kramer, Martin, Reiniger, Smith, and Sullivan [73] in the induced case.

Spreadness In the previous paragraph, we discussed the task of designing networks under constraints. Once a proposed network has been constructed, a natural next step is to assess its *robustness*. The concept of robustness is difficult to formalise precisely; however, one intuitive approach is to analyze how the network would behave when some communication links are destroyed and whether the original required structure would remain. *Spreadness* is a mathematical tool that allows us to study this phenomenon. Simply put, proving that a substructure admits a *spread embedding* ensures that random disturbances to the network are highly unlikely to compromise the substructure in question.

In addition to the motivation presented above, our study is rooted in a well-established area of extremal combinatorics: the investigation of minimum-degree thresholds for graph properties. This research theme can be traced back to Dirac's seminal result in 1952 [63], which states that any n-vertex graph with a minimum degree of at least $n/2$ is Hamiltonian. Over the past seventy years, researchers have made significant efforts to strengthen this theorem, leading to two main research directions: *enumeration* and *robustness* results. Enumeration results provide lower bounds on the number of copies of the target structure [165, 58], while robustness results, as discussed above, investigate how many edges can be removed from the host graph while still preserving the target property [125, 174].

The recent breakthrough proof of the Kahn-Kalai conjecture [149] has sparked significant interest in developing "spread" versions of Dirac-type theorems [14, 101, 105, 114, 153, 30], as these would directly lead to enumeration and robustness results, merging two research threads that have largely been explored separately until now. In this manuscript we will study spread embedding for bounded degree trees.

Universality During the rapid expansion of the processing chip market, some manufacturers aimed to design configurable chips—single chips that could be adapted into multiple different configurations by selectively removing connectors or components. To achieve this, they designed what mathematicians and computer scientists call a *universal* graph: a graph that contains every member of a targeted family of graphs as a subgraph. The problem of designing configurable chips has drawn significant attention to universal structures within the computer science community over the past few decades. However, such structures have

been extensively studied by mathematicians for more than seventy years [75, 99, 155][3]. In particular, the search for universal posets played a foundational role in the development of category theory [92]. In graph theory, the Rado graph, the first discovered universal structure for the class of all countable graphs [155], has led to numerous discoveries in multiple fields of combinatorics, as explained in the following surveys [43, 44].

In the context of chip design, the configurable chip has the same physical constraints as the different chips it is supposed to emulate. In graph-theoretic terms, this implies that the underlying (universal) graph should be what we will refer to in this manuscript as "faithful," meaning that the universal graph itself belongs to the class it aims to represent. For instance, if all graphs underlying the different chips must be planar, then the configurable chip's graph must also be planar. Surprisingly, the study of "faithful" universal graphs for infinite countable structures has a rich history of important results [148, 93] in mathematics[3]. On the contrary, the main techniques developed by mathematicians and computer scientists for constructing finite universal graphs often yield graphs that are far from faithful. In Chapter 5, we investigate the problem of designing faithful or near-faithful structures in a finite setting.

[3] See Section 5.1 for additional references.

Introduction étendue en français

Un *graphe* G est constitué d'un ensemble d'objets abstraits $V(G)$ appelés *sommets* et d'un ensemble de "connexions" entre ces sommets, noté $E(G)$, qui représentent des paires d'éléments de $V(G)$. Cette structure mathématique simple permet de modéliser une grande variété de situations réelles. Par exemple, on peut considérer le graphe G associé à un réseau social, où les sommets représentent les personnes inscrites et les arêtes représentent la relation "d'ami". Une fois cet objet mathématique construit, on peut traduire des questions concrètes sur le réseau social dans le language de la théorie des graphes. Supposons que vous vouliez savoir qui forme votre plus grand groupe d'amis. Cela reviendrait à identifier le plus grand ensemble de sommets de G, tous reliés entre eux et incluant le sommet qui vous représente.

Mais la polyvalence des graphes va bien au-delà de leur utilisation pour représenter des réseaux sociaux. Le compilateur LaTeXque j'utilise en ce moment pour rédiger ce manuscrit s'appuie sur la theory des graphes pour allouer des variables aux registres de mon ordinateur. L'itinéraire que je prendrai pour rentrer chez moi ce soir, après une journée productive passée à l'écriture de cette thèse, a été conçu grâce à la théorie des graphes. Et lorsque je serai enfin chez moi, que j'essayerai de me détendre devant un film sur ma plateforme de streaming préférée, ce sera encore un algorithme optimisé grâce à la théorie des graphes qui déterminera les recommendations qui me seront proposées.

La théorie des graphes est l'une des branches les plus anciennes de la *combinatoire*, elle-même une branche des mathématiques. Bien qu'il n'existe pas de définition universellement acceptée de la combinatoire, nous la décrirons ici comme l'étude des structures finies à travers leur énumération, leur construction, leur optimisation et leur analyse. Ce manuscrit explore divers problèmes en combinatoire, le plus souvent liés à la théorie des graphes. Commençons par présenter les quatre notions qui forment le titre de cette thèse : Reconstruction, Saturation, Dispersion et Universalité.

Reconstruction Vous êtes-vous déjà demandé quelles sont les meilleures questions que l'on puisse poser durant une partie du jeu de société *Qui est-ce ?* ? Pour ceux qui n'ont pas eu la chance de jouer à ce classique de l'enfance, *Qui est-ce ?* est un jeu à deux joueurs où chacun pose à tour de rôle des questions fermées (oui/non) pour deviner le personnage choisi par son adversaire. Ce jeu illustre parfaitement un problème de reconstruction en combinatoire. On commence avec un grand ensemble de personnages, dont l'un a été secrètement, voire malicieusement, sélectionné par notre adversaire. L'objectif est de déterminer, ou en termes mathématiques, reconstruire, quel personnage a été choisi. Pour y parvenir, nous ne pouvons poser que des questions fermées—requêtes—qui fournissent des indices partiels sur l'identité du personnage. La difficulté réside dans le fait que notre adversaire poursuit le

même objectif, et que nous sommes engagés dans une course pour identifier le personnage adversaire en premier. Chaque question ayant un coût, le but est de découvrir la réponse avant notre adversaire, et donc en posant le moins de questions possible.

Si vous êtes mathématicien — ou simplement si vous êtes quelqu'un qui aime gâcher une partie de jeu de société en utilisant de la logique — vous pourriez élaborer une stratégie optimale pour *Qui est-ce ?*, garantissant que, si le jeu commence avec n personnages possibles, vous pourrez toujours deviner celui de votre adversaire en au plus $\lceil \log_2 n \rceil$ questions. De plus, ce résultat est optimal : si vous posez strictement moins de questions, il pourrait exister deux personnages indistinguable.

Les questions de reconstruction sont généralement classées en deux catégories : *existence* et *optimisation*. Les questions d'*existence* cherchent à savoir si un objet peut être reconstruit de manière unique, même avec un nombre illimité de requêtes. Si la réponse à la question d'*existence* est positive, la question naturelle qui suit est une question d'*optimisation* : quel est le nombre minimal de requêtes nécessaires pour déterminer l'objet de manière unique ?

L'un des problèmes les plus anciens et les plus célèbres de reconstruction de graphes est une question d'*existence*, souvent appelée la *Graph Reconstruction Conjecture*, formulée par Stanisław-Ulam vers 1941. Elle demande si tout graphe simple et fini G ayant au moins trois sommets peut être déterminé de manière unique (à isomorphisme près) à partir de son "*deck*", défini comme la multiensembles des sous-graphes obtenus en retirant un sommet v de G pour chaque $v \in V(G)$. Malgré des efforts considérables de la communauté scientifique et des réponses positives pour certaines classes de graphes, le problème reste ouvert.

Un autre problème fondamental, appartenant à la catégorie des questions d'*optimisation*, est l'*Evasivness Conjecture*, aussi connue comme conjecture d'Aanderaa–Karp-Rosenberg, du nom des trois chercheurs l'ayant introduite autour de 1973 [163]. Ce problème dépasse le cadre de la théorie des graphes, mais nous nous concentrerons ici sur sa formulation en termes de graphes. On cherche à reconstruire l'ensemble des arêtes d'un graphe G, en utilisant uniquement des questions appelées *requêtes d'arêtes*, sont de la forme : étant donnés $u, v \in V(G)$, "Existe-t-il une arête entre u et v dans G ?" Une classe de graphes $\mathcal{G}$ est dite *monotone* si, pour tout $G \in \mathcal{G}$, tout graphe H obtenu en ajoutant des arêtes à G appartient aussi à $\mathcal{G}$. L'*Evasivness Conjecture* affirme que, pour toute classe de graphes $\mathcal{G}$ sur n sommets qui n'est ni vide ni triviale[4], déterminer si un graphe G appartient à $\mathcal{G}$ nécessite $\binom{n}{2}$ requêtes d'arêtes dans le pire des cas. Intuitivement, cela signifie qu'un algorithme doit vérifier *toutes* les paires possibles de sommets avant de pouvoir décider si $G \in \mathcal{G}$. La conjecture reste ouverte à ce jour et n'a été démontrée que lorsque n premier par Rivest et Vuillemin en 1976.

[4] On exclut la classe vide et la classe contenant tous les graphes.

Saturation Supposons que vous deviez concevoir un réseau informatique pour une entreprise en respectant deux exigences. La première est d'utiliser le moins de connexions possible, et la seconde est que certains ordinateurs ne doivent pas pouvoir communiquer directement pour des raisons de sécurité. Si nous représentons les ordinateurs par des sommets d'un graphe et les connexions par des arêtes, la demande de l'entreprise revient à résoudre un problème de saturation. De manière plus générale, un problème de saturation consiste à trouver une structure combinatoire qui maximise ou minimise une fonction de coût globale — dans notre exemple, le nombre de connexions — tout en respectant des contraintes locales, ici les connexions interdites.

De nombreux problèmes fondamentaux en combinatoire ont été ou peuvent être formulés sous forme de problèmes de saturation. Par exemple, en théorie structurelle des graphes, les graphes *critiques*, introduits vers 1950 par G. A. Dirac, sont des structures saturées qui ont été largement étudiées[5] afin de mieux comprendre le comportement du nombre chromatique. En combinatoire extrémale, plusieurs résultats fondamentaux découlent de l'étude des graphes H-saturés, dont l'histoire remonte à plus d'un siècle, en commençant par les travaux de Mantel sur les graphes sans triangles [135], et se poursuivant avec le célèbre théorème de Turán [178] et tout autant célèbre théorème d'Erdős, Hajnal et Moon [68]. Un graphe H-saturé est un graphe G qui ne contient pas H comme sous-graphe, mais où l'ajout de toute arête à G entraîne l'apparition d'une copie de H. Grâce aux deux théorèmes mentionnés ci-dessus et aux efforts considérables de la communauté scientifique — voir le "dynamic survey" de Faudree, Faudree et Schmitt [71] — nous disposons aujourd'hui d'une compréhension approfondie du nombre d'arêtes dans les graphes H-saturés.

Récemment, les chercheurs ont commencé à explorer des variantes et des généralisations des graphes H-saturés. En particulier, il existe un intérêt croissant pour l'étude des graphes *H-induit-saturés*, où la contrainte d'interdire H comme sous-graphe est remplacée par l'interdiction de H comme sous-graphe induit. Cette modification rend le problème nettement plus complexe, et pour la plupart des graphes, l'existence même de telles structures saturées reste inconnue [137, 32, 16].

Dans cette thèse, nous étudions la saturation dans les ensembles ordonnés partiellement. Étant donné un poset P, nous cherchons à déterminer la taille minimale d'une famille P-saturée $\mathcal{F} \subseteq 2^{[n]}$, définie comme une famille $\mathcal{F}$ qui ne contient pas P comme sous-poset induit, mais telle que pour tout $X \in 2^{[n]} \setminus \mathcal{F}$, la famille $\mathcal{F} \cup X$ en contienne une copie induite de P. Cet axe de recherche a été initié pare un résultat datant de 1981 de Katona et Tarjan [112] et a été ensuite formalisée dans la terminologie que l'on utilise aujourd'hui par Gerbner, Keszegh, Lemons, Palmer, Pálvölgyi et Patkós [82] pour la version non induite, puis par Ferrara, Kay, Kramer, Martin, Reiniger, Smith et Sullivan [73] dans le cas induit.

[5] Nous renvoyons le lecteur au livre récent de Stiebitz, Schweser et Toft [173] pour un aperçu du sujet.

Dispersion Dans le paragraphe précédent, nous avons utilisé l'exemple de la conception d'un réseaux de communication sous contraintes de sécurtiés. Une fois un tel réseau construit, l'étape suivante naturelle consiste à évaluer sa *robustesse*. La robustesse est un concept difficile à formaliser précisément ; cependant, intuitivement, analyser le comportement du réseau en cas de destruction de certaines connexions et à vérifier si la structure requise initialement est préservée, permet de juger de sa robustesse. La notion de *dispersion* est un outil mathématique permettant d'étudier ce phénomène. Plus précisément, prouver qu'une sous-structure admet un *plongement dispersé* garantit que des perturbations aléatoires du réseau ont très peu de chances de compromettre la sous-structure en question.

En plus de la motivation présentée ci-dessus, l'étude réalisé dans cette thèse s'inscrit dans une branche bien établi de la combinatoire extrémale : l'étude des seuils de degré minimal assurant certaines propriétés des graphes. Ce thème de recherche remonte au résultat de Dirac en 1952 [63], qui établit que tout graphe à n sommets dont le degré minimal est au moins $n/2$ est hamiltonien. Depuis plus de soixante-dix ans, de nombreux chercheurs ont cherché à enforcer et généraliser ce théorème, menant à deux grandes directions de recherche : les résultats d'*énumération* et de *robustesse*. Les résultats d'énumération fournissent des bornes inférieures sur le nombre de copies de la structure cible [165, 58], tandis que les résultats de robustesse, comme mentionné ci-dessus, étudient combien d'arêtes peuvent être supprimées du graphe tout en préservant la propriété étudiée [125, 174].

L'avancement récent sur la conjecture de Kahn-Kalai [149] a suscité un intérêt croissant pour le développement de versions "dispersées" des théorèmes de type Dirac [14, 101, 105, 114, 153, 30], car ces résultats permettraient d'obtenir simultanément des conclusions d'énumération et de robustesse, réunissant ainsi deux axes de recherche jusqu'ici largement explorés séparément. Dans ce manuscrit, nous étudions les plongements dispersés dans les graphes denses, pour les arbres de degré borné.

Universalité Lors de l'essor rapide du marché des processeurs, certains fabricants ont cherché à concevoir des puces reconfigurables — des circuits uniques pouvant être adaptés à différentes configurations en supprimant sélectivement certains connecteurs ou composants. Pour cela, ils ont conçu ce que les mathématiciens et informaticiens appellent un graphe *universel* : un graphe qui contient chaque élément d'une famille cible comme sous-graphe. Le problème de la conception de puces configurables a suscité un grand intérêt pour les structures universelles dans la communauté informatique au cours des dernières décennies. Cependant, de telles structures ont été étudiées par les mathématiciens depuis plus de soixante-dix ans [75, 99, 155][6]. En particulier, la recherche de posets universels a joué un rôle fondateur dans le développement de la théorie des catégories [92]. En théorie des graphes, le graphe de Rado, l'une des premières structures universelles découvertes pour la

[6]Voir Section 5.1 pour des références supplémentaires.

classe de tous les graphes dénombrables, a conduit à de nombreuses avancées dans plusieurs domaines de la combinatoire, comme expliqué dans les *surveys* suivants [43, 44].

Dans le cadre de la conception de puces, la puce reconfigurable est soumise aux mêmes contraintes physiques que les puces qu'elle est censée émuler. En termes de théorie des graphes, cela signifie que le graphe sous-jacent (universel) doit être ce que nous appellerons dans ce manuscrit un graphe "fidèle", c'est-à-dire un graphe appartenant lui-même à la classe qu'il vise à représenter. Par exemple, si toutes les puces doivent être basées sur des graphes planaires, alors le graphe de la puce reconfigurable doit aussi être planaire. De manière surprenante, l'étude des graphes universels fidèles pour les structures infinies dénombrables possède une riche histoire et plusieurs résultats importants [148, 93][6] en mathématiques. En revanche, les principales techniques développées par les mathématiciens et informaticiens pour la construction de graphes universels finis produisent souvent des graphes très éloignés d'une structure fidèle. Dans Chapter 5, nous étudions le problème de la conception de structures fidèles ou quasi-fidèles dans un cadre fini.

Preliminaries

This section outlines the basic notations and tools utilised throughout this manuscript, most of which are standard within the fields of combinatorics and graph theory.

Graphs and operations

For $n \in \mathbb{N}^*$, let $[n]$ denote the set $\{1, 2, \dots, n\}$. For a set A let 2^A denote the power set of A and, for $m \in \mathbb{N}$, let $\binom{A}{m}$ denote the set $\left\{a \in 2^A \,\middle|\, |a| = m\right\}$. Similarly $\binom{A}{\leqslant m} = \left\{a \in 2^A \,\middle|\, |a| \leqslant m\right\}$.

A *graph* $G = (V(G), E(G))$ consists of a set of vertices $V(G)$ and a set of edges $E(G) \subseteq \binom{V(G)}{2}$. For two distinct vertices $u, v \in V(G)$, we often write the edge $\{u, v\}$ as uv. If $uv \in E(G)$, we say that u and v are *neighbours*. The *open neighbourhood* of a vertex v, denoted $N(v)$, is the set of all neighbours of v, and its size $|N(v)|$ is called the *degree* of v. The *closed neighbourhood*, denoted $N[v]$, is defined as $N[v] := N(v) \cup \{v\}$.

A *path* of length ℓ between vertices u and v in G is a sequence of distinct vertices $u, w_1, w_2, \dots, w_{\ell-1}, v$ such that $uw_1, w_1w_2, \dots, w_{\ell-2}w_{\ell-1}, w_{\ell-1}v \in E(G)$. The *distance* between two vertices u, v, denoted $d_G(u, v)$, is defined as:

- $d_G(u, v) = 0$ if $u = v$,
- The length of a shortest path from u to v if such a path exists,
- $d_G(u, v) = +\infty$ if no path exists between u and v.

A graph is called *connected* if $d_G(u, v) < \infty$ for all $u, v \in V(G)$ and *disconnected* otherwise.

For $k \geqslant 1$, the *k-neighbourhood* of a vertex v is defined as

$$N^k(v) = \{u \in V(G) \mid d_G(u, v) = k\},$$

and the *ball of radius k* around v is defined as

$$N^{\leqslant k}(v) = \{u \in V(G) \mid d_G(u, v) \leqslant k\}.$$

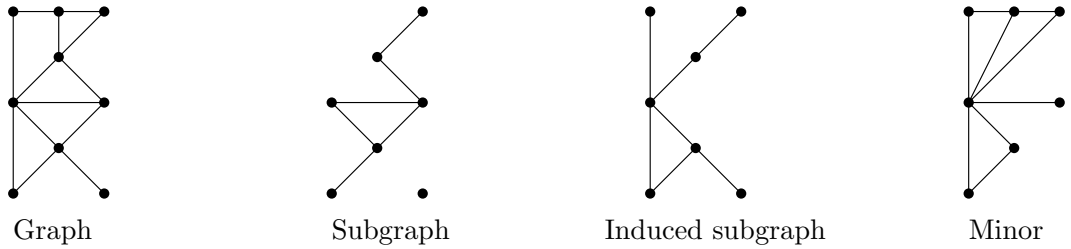


Figure 1: Example of a graph (left) and substructures of this graph.

Given a graph $G = (V(G), E(G))$, H is called a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. The graph H is called an induced subgraph if for any two vertices $u, v \in V(H)$, $uv \in E(H) \Leftrightarrow uv \in E(G)$. Note that an induced subgraph H of G is completely defined by G and the subset of vertices $V(H) \subseteq V(G)$, in particular, we can define the graph induced on some vertex set $S \subseteq V(G)$, denoted $G[S]$ as the only induced subgraph of G which satisfies $V(H) = S$. Similarly, we define $G \setminus S := G[V(G) \setminus S]$. Given two graphs G_0 and G_1 the disjoint union of G and H is the graph $G_0 + G_1 = (\{(0, u) \mid u \in V(G)\} \cup \{(1, u) \mid u \in V(H)\}, \{(i, u)(i, v) \mid uv \in E(G_i)\})$.

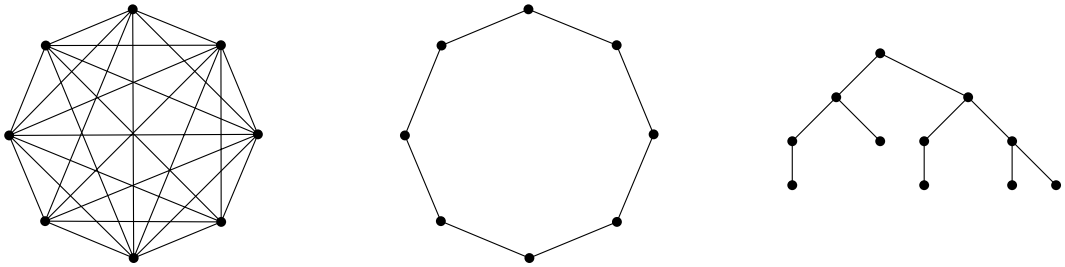
A minor of a graph G is a graph obtained by deleting some vertices and edges, and contracting edges. The contraction operation of $uv \in E(G)$ consists in replacing the vertices u and v by a new vertex with neighbourhood $N(u) \cup N(v)$. See Fig. 1 for an example of the different graph operations described.

For some recurring graphs, we introduce names and notation. The graph consisting only of a path of length n, is denoted P_n. The complete graph or clique on n vertices, denoted K_n, is the only graph containing all $\binom{n}{2}$ possible edges. The cycle on n vertices, denoted C_n, is the graph obtained from a path P_n defined by the sequence $u_1, \ldots, u_n$ when we add the edge $u_n u_1$. A tree is a connected graph that does not contain any cycle as an induced subgraph, and the disjoint union of an arbitrary number of trees is called a forest. See Fig. 2.

Algorithmic complexity

When studying complexity, or asymptotic behaviours, we often use the Bachmann-Landau notation, also known as asymptotic notation: $o, O, \Theta, \sim, \Omega, \omega$. Given a parameter $\Delta \in \mathbb{N}$, we write $f(n, \Delta) = o_\Delta(g(n))$ if for every fixed $\Delta \in \mathbb{N}$, $f(n, \Delta) = o(g(n))$ as $n \to +\infty$. We define similarly $O_\Delta, \Theta_\Delta, \sim_\Delta, \Omega_\Delta, \omega_\Delta$ and we omit the assumption on the growth of the asymptotic variable, here "$n \to +\infty$", when clear from the context.

When studying randomised algorithms, the randomised query complexity is defined as the maximum, taken over all hidden graphs G, of the expected number of queries made by



K_9 C_9 A tree

Figure 2: From left to right, a representation of K_9, C_9, and a tree.

the algorithm. We sometimes write that an event $E(n)$, which depends on some parameter n holds, *with high probability*, abbreviated to w.h.p., when $E(n) = 1 - o(1)$ when $n \to +\infty$.

Throughout this manuscript, we use standard notation for probability. We refer the reader to [37] for further details on these notations and their standard applications in random graphs. The following concentration bound, due to Chernoff, is particularly useful throughout this work.

Lemma 0.1 (Chernoff's bound [52]). *Given n independent random variables $X_1, \dots, X_n$ taking values in $\{0, 1\}$, let us denote by $\mu = \mathbb{E}[\sum_{i=1}^{n} X_i]$. Then for any $0 < \delta < 1$,*

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} X_i - \mu\right| \geqslant \delta\mu\right) \leqslant 2e^{-\delta^2\mu/3}.$$

Also for any $\delta \geqslant 0$,

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \geqslant (1+\delta)\mu\right) \leqslant e^{-\delta^2\mu/(2+\delta)}.$$

Graph parameters

In the early 1980s Robertson and Seymour revolutionised structural graph theory. As part of their Graph Minors project, which consists of more than 20 papers published between 1983 [160] and 2010 [159], they established the concept of treewidth [158] to characterise how "tree-like" a graph is, providing a measure of its structural complexity.

We will now give the definition of different, extensively studied, graph parameters which are mentioned in this manuscript. When these parameters form an important part of a statement or a proof mentioned in this manuscript, they will also be recalled in the specific preliminaries of the chapter.

A *tree decomposition* of a graph G is a tuple $(T, (B_t)_{t\in V(T)})$ where T is a tree and B_t is a subset of $V(G)$ for every $t \in V(T)$, for which the following conditions hold.

- For every $v \in V(G)$, the set of $t \in V(T)$ such that $v \in B_t$, is non-empty and induces a subtree of T.
- For every $uv \in E(G)$, there exists a $t \in V(T)$ such that $\{u, v\} \subseteq B_t$.

We say that a tree decomposition is trivial if T has a single vertex. The *width* of a tree decomposition $(T, (B_t)_{t\in V(T)})$ is the maximum of $|B_t|$ over $t \in V(T)$.

The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum over all tree decompositions $(T, (B_t)_{t\in V(T)})$ of G, of the width of $(T, (B_t)_{t\in V(T)})$.

The *pathwidth* of a graph G denoted $\mathrm{pw(G)}$, is the minimum over all tree decompositions $(P, (B_t)_{t\in V(P)})$ of G where P is restricted to be a path, of the width of $(P, (B_t)_{t\in V(P)})$.

The *treelength* of a graph G (denoted $\mathrm{tl}(G)$) is the minimal integer k for which there exists a tree decomposition $(T, (B_t)_{t\in V(T)})$ of G such that $d_G(u, v) \leqslant k$ for every pair of vertices u, v that share a bag (i.e. $u, v \in B_t$ for some $t \in V(T)$).

The *treedepth* of a graph G, denoted $\mathrm{td}(G)$ can be defined recursively as follows,

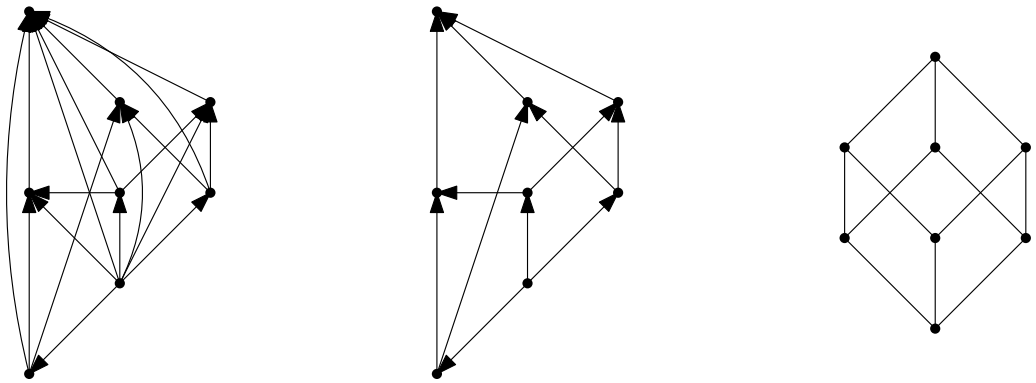
$$\mathrm{td}(G) = \begin{cases} 1, & \text{if } |G| = 1; \\ 1 + \min_{v\in V(G)} \mathrm{td}(G \setminus \{v\}), & \text{if } G \text{ is connected and } |G| \geqslant 2; \\ \max_i \mathrm{td(G_i)}, & \text{otherwise, for } \{G_i\}_i \text{ the set of connected components of } G. \end{cases}$$

Finally, the *bandwidth* of a n-vertex graph G, is the minimum over all total orders on $V(G) = \{v_1, \ldots, v_n\}$ of the maximum of $|i - j|$ for $v_iv_j \in E(G)$.

Partially ordered sets

In Chapter 3 and Chapter 5, we study *partially ordered sets*, also referred to as *posets*. They are defined as a pair $(P, \preccurlyeq)$, where P is a set of elements and $\preccurlyeq$ is a partial order on P, meaning $\preccurlyeq$ is a binary relation over P satisfying the following three properties: for any three elements $a, b, c \in P$:

- **Reflexivity**: $a \preccurlyeq a$.
- **Antisymmetry**: If $a \preccurlyeq b$ and $b \preccurlyeq a$ then $a = b$
- **Transitivity**: If $a \preccurlyeq b$ and $b \preccurlyeq c$ then $a \preccurlyeq c$.



comparability graph cover graph Hasse diagram

Figure 3: Different representation of $(2^{[3]}, \subseteq)$, from left to right, its comparability graph, cover graph and Hasse diagram.

Similarly to $\leqslant$ and $<$, we denote by $\prec$ the strict partial order relation associated with $\preccurlyeq$. Formally, we define $\prec = \preccurlyeq \setminus I_P$, where $I_P = \{(p, p) \mid p \in P\}$ represents the identity relation.

A poset can be seen as a special case of a graph, where we add an orientation to the edges. Given a poset $(P, \preccurlyeq)$, the *comparability graph* of P is $G = (V(G), E(G))$, where $V(G) = P$ is the set of vertices of G, and $E(G) \subseteq V \times V$ represents the oriented edges of G. These edges are given by $E(G) = \{(a, b) \in P \times P \mid a \preccurlyeq b\}$. We also define another oriented graph called the *cover graph* of $(P, \preccurlyeq)$ as the graph $H = (V(H), E(H))$ with $V(H) = P$ and $E(H)$ represents the "immediate relation" of $\preccurlyeq$, that is the relation $a \preccurlyeq b$ such that there is no $c \in P$ with $a \preccurlyeq c \preccurlyeq b$, formally $E(H) = \{(a, b) \in P \times P \mid a \preccurlyeq b \wedge (\forall c \in P, a \not\preccurlyeq c \vee c \not\preccurlyeq b)\}$. We often use the *cover graph* to represent a poset P. This representation, called the *Hasse diagram* of P, is obtained by drawing the cover graph of P such that each edge is oriented from the lower to the upper vertex. See Fig. 3.

One of the most studied posets, also central to some results presented in this thesis, is the *hypercube*, also called the *Boolean lattice*. The hypercube of dimension n, denoted $(2^{[n]}, \subseteq)$ and sometimes simply written as Q_n, consists of the power set of $[n]$ ordered by inclusion. For any $i \in [n]$, the layer i of $2^{[n]}$ is denoted $\binom{[n]}{i}$.

A *chain* $C \subseteq 2^{[n]}$ is a set system consisting of pairwise comparable elements,i.e., $X \subseteq Y$ or $Y \subseteq X$ for all $X, Y \in C$. An *antichain* is a set system consisting of elements that are pairwise incomparable.

We say a chain C in $2^{[n]}$ *starts* in R and *ends* in S if the smallest element of C is in R and the largest element of C is in S. We say a chain $C_1 \subseteq \cdots \subseteq C_m$ is *skipless* if $|C_{i+1}| = |C_i| + 1$ for all $i \in [m-1]$ i.e. the chain does not 'skip' over any layers.

A *chain decomposition* of a set system $\mathcal{F} \subseteq 2^{[n]}$ is a collection of disjoint chains $C^1, \ldots, C^m \subseteq \mathcal{F}$ such that $\mathcal{F} = \cup_{i=1}^{m} C^i$, that is, for each $X \in \mathcal{F}$, there is exactly one $i \in [m]$ such that C^i contains the set X. The size of the chain decomposition is the number of chains m. The following theorem, known as Dilworth's theorem, is a fundamental result on posets.

Theorem 0.2 (Dilworth [61]). *Let n be an integer and $\mathcal{F} \subseteq 2^{[n]}$, then the size of the largest antichain in $\mathcal{F}$ is equal to the minimum size of a chain decomposition of $\mathcal{F}$.*

Another well-known structural result in the area, often referred to as the symmetric chain decomposition (see [6, 86]) can be stated as follows.

Lemma 0.3. *There is a skipless chain decomposition of $2^{[n]}$ into $\binom{[n]}{\lfloor n/2 \rfloor}$ chains. In particular, there is a matching of size $\binom{n}{s}$ from $\binom{[n]}{s}$ to $\binom{[n]}{r}$ whenever $s < r \leqslant \lceil n/2 \rceil$ or $s > r \geqslant \lfloor n/2 \rfloor$.*

Chapter 1

Overview of the results

This chapter summarises the main results presented in this thesis. It is intended for readers familiar with combinatorial and graph theory notions. Further formal definitions are given in each of the relevant chapters. It should be considered a roadmap of the thesis rather than an introduction to the cited results.

1.1 Distance reconstruction

In Chapter 2, we study the following reconstruction problem: our input is a graph $G = (V, E)$ where V is known and E is hidden. We are given access to an oracle that takes as input two vertices $u, v \in V$ and outputs $d_G(u, v)$. The goal is to reconstruct the edge set of G completely. We say that an algorithm reconstructs a class of graphs $\mathcal{G}$ if the output of the sequence of queries made by the algorithm uniquely identifies $G \in \mathcal{G}$.

In joint work with Carla Groenland we proved the following three theorems, presented in Section 2.3.

Theorem 2.8 ([25]). *There exists a deterministic algorithm to reconstruct the class of connected n-vertex chordal graphs of maximum degree at most $\Delta \leqslant \sqrt{\log n}$ using $O(\Delta n \log n)$ distance queries.*

We extend the previous algorithm to k-chordal graphs, i.e., graphs that exclude C_r as an induced subgraph for every $r \geqslant k$.

Theorem 2.9 ([25]). *There exists a deterministic algorithm to reconstruct the class of connected k-chordal graphs of maximum degree at most Δ on n vertices using $O_{\Delta,k}(n \log n)$ queries.*

For the more restricted class of bounded degree trees, we obtain the following result.

Theorem 2.10 ([25]). *There exists a deterministic algorithm to reconstruct the class of trees of maximum degree at most Δ on n vertices using $\Delta n \log_\Delta n + (\Delta + 2)n$ queries.*

We also improved the best-known lower bound for distance reconstruction of bounded degree graphs. The following theorem is the main consequence of our work. In particular, it implies that the three theorems above have an optimal dependency on n, and Theorem 2.10 also has an optimal dependency on Δ; it is therefore optimal up to a $O(1)$-factor. Nonetheless, in Section 2.5, we developed a more general statement, applicable to a broad range of reconstruction setups.

Theorem 2.14 ([25]). *Let $\Delta \geqslant 2$ and $n = 2c\Delta^k$ be integers, where $c \in [1, \Delta)$ and $k \geqslant 50(c \ln c + 3)$ is an integer. Then, any randomised algorithm requires at least $\frac{1}{50}\Delta n \log_\Delta n$ queries in expectation to reconstruct n-vertex trees of maximum degree $\Delta + 1$.*

In a sole-author paper, I studied the average complexity of reconstructing a graph sampled from a distribution $\mathcal{D}$ over the set of graphs on n vertices. In this case, a graph $G \sim \mathcal{D}$ is said to be reconstructed if the sequence of queries done by the algorithm uniquely identifies G, not only among a family of graphs $\mathcal{G}$, but also among all graphs on n vertices. The query complexity is computed as the expected number of queries made by the algorithm when the input G is sampled from the distribution $\mathcal{D}$.

Theorem 2.12 ([19]). *For any $\varepsilon \geqslant 0$, and every $n \in \mathbb{N}$, for every $\frac{2000 \log n}{n} \leqslant p \leqslant n^{-\frac{1}{2}-\varepsilon}$, there exists an algorithm that reconstructs $G \sim G(n,p)$ using $O(\Delta^2 n \log n)$ queries in expectation, where $\Delta = (n-1)p$ is the expected average degree of G.*

1.2 Poset saturation

In Chapter 3, we study the problem of saturation in the hypercube. Let us denote by $2^{[n]}$ the set containing all subsets of $\{1, \dots, n\}$. Formally, a family $\mathcal{F} \subseteq 2^{[n]}$ is called P-saturated in $2^{[n]}$ if $\mathcal{F}$ satisfies the following two conditions:

- P is not an induced subposet of $\mathcal{F}$,
- for any $x \in 2^{[n]} \setminus \mathcal{F}$, $\mathcal{F} \cup \{x\}$ contains P as an induced subposet.

Given a poset P and an integer n, we are interested in the saturation number of P, denoted $\mathrm{sat}^*(n, P)$, which is defined as the minimum size of a P-saturated family in $2^{[n]}$.

In joint work with Carla Groenland, Hugo Jacob, and Tom Johnston [28], we managed to pinpoint the saturation number of the antichain of size k (denoted A_k). This is the first

infinite family of posets for which the saturation number is computed exactly, and it answers positively to two conjectures [73, 15]. Our proof proceeds via a generalisation of a theorem from Lehman and Ron [128], which is of independent interest. These results are presented in Section 3.4.

A chain $C_1 \subseteq C_2 \subseteq \ldots \subseteq C_r \subseteq [n]$ is *skipless* if it satisfies the property $|C_{i+1}| = |C_i| + 1$ for all $i \in [r-1]$.

Theorem 3.10 ([28]). *Suppose that $\mathcal{F} \subseteq 2^{[n]}$ admits a chain decomposition into m chains. Then there exist disjoint skipless chains $C^1, \ldots, C^m \subseteq 2^{[n]}$ such that $\mathcal{F} \subseteq \bigcup_{i=1}^{m} C^i$.*

The following corollary, already answering both conjectures, can be derived from Theorem 3.10.

Corollary 3.11 ([28]). *There exist constants $c_1, c_2 > 0$ such that for all $k \geqslant 4$ and n sufficiently large,*

$$n(k-1) - c_1 k \log k \leqslant \mathrm{sat}^*(n,k) \leqslant n(k-1) - c_2 k \log k.$$

With substantially more effort it is possible to pinpoint precisely the value of $\mathrm{sat}^*(n, A_k)$. We obtain the following theorem.

Theorem 3.8 ([28]). *Let $n, k \geqslant 4$ be integers and let ℓ and $c_0, \ldots, c_{\lfloor \ell/2 \rfloor}$ be as defined above. If $n < \ell$, then $\mathrm{sat}^*(n,k) = 2^n$. If $n \geqslant \ell$, then*

$$\mathrm{sat}^*(n,k) \geqslant 2\sum_{t=0}^{\lfloor \ell/2 \rfloor} c_t + (k-1)(n-1-2\lfloor \ell/2 \rfloor).$$

Moreover, equality holds when $n \geqslant 2\ell + 1$.

By combining a published joint work with Carla Groenland, Maria-Romina Ivan and Tom Johnston [27], as well as an ongoing joint work with Jędrzej Hodor, Hoang La and William T. Trotter, we obtain the first general bounds for the function sat^*. Little is known about the possible behavior of this function. Prior to our work, no non-trivial upper bounds were known.

Theorem 3.15 ([27]). *For any fixed poset P and any $n \in \mathbb{N}$, $\mathrm{sat}^*(n,P) = O(n^c)$, where $c \leqslant |P|$ is a constant depending on P only.*

To prove Theorem 3.15, we introduce two new key notions, "cube-height" and "cube-width". Intuitively, for a poset P, the "cube-height" is the least k such that, for some n, we can embed P into the first $k+1$ layers of Q_n, while the "cube-width" is the smallest n that makes such a "small height" embedding possible.

Definition. *For a poset P, the cube-height $\mathrm{h}^*(P)$ is the minimum $h^* \in \mathbb{N}$ for which there exists $n \in \mathbb{N}$ such that $\binom{[n]}{\leqslant h^*}$ contains an induced copy of P.*

Definition. *For a poset P, the cube-width $\mathrm{w}^*(P)$ is the minimum $w^* \in \mathbb{N}$ such that there exists an induced copy of P in $\binom{[w^*]}{\leqslant \mathrm{h}^*(P)}$.*

In the proof of Theorem 3.15 we bound this constant c by $\mathrm{w}^*(P)$, to obtain the result as stated, we also prove the following theorem.

Theorem 3.19. *For any poset P, $\mathrm{w}^*(P) \leqslant |P|$.*

1.3 Spread embedding of trees in dense graphs

In Chapter 4 we study spreads embedding. Given two graphs G and H an embedding of H in G is an injection $\varphi : V(H) \to E(G)$ such that for any $uv \in E(H)$, $\varphi(u)\varphi(v) \in E(G)$.

Definition ([154]). *Let G and H be finite graphs, and let μ be a probability distribution over embeddings of H into G. For $q \in [0,1]$, we say that μ is q-spread if for every two sequences of distinct vertices $x_1, \dots, x_s \in X$ and $y_1, \dots, y_s \in Y$,*

$$\mu\left(\{\varphi : \varphi(x_i) = y_i \text{ for all } i \in [s]\}\right) \leqslant q^s.$$

The concept of spreadness has been recently introduced in order to generalise *robustness* and *counting* results on Dirac's thresholds, more details about this motivation is given in Section 4.1. The main result of Chapter 4, joint work with Clément Legrand-Duchesne and Alp Müyesser [30] is a new proof of the following theorem.

Theorem 4.3 (Pham, Sah, Sahwney, Simkin [154, 30]). *For every $\Delta \in \mathbb{N}$ and $\alpha > 0$, there exists $n_0, C \in \mathbb{N}$ such that the following holds for all $n \geqslant n_0$. If G is an n-vertex graph with $\delta(G) \geqslant (1+\alpha)\frac{n}{2}$, and T is a n-vertex tree with $\Delta(T) \leqslant \Delta$, there exists a (C/n)-spread distribution on embeddings of T onto G.*

Contrary to the original proof by Pham, Sah, Sahwney and Simkin [154], the proof presented in this manuscript avoids the use of Szemerédi's regularity lemma, leading to improved constants. The newly developed techniques are flexible and may be of interest for future research on spread embeddings.

1.4 Faithful universal graphs and posets

In Chapter 5, we investigate "faithful" universal structures, which can be intuitively understood as universal structures that satisfy the properties of the class they aim to represent. In the infinite setting, such structures have been studied for several decades[1]. For instance, the question of whether there exists a countable planar graph that contains every countable planar graph was posed by Ulam and resolved by Pach in 1981 [148]. This result was recently strengthened by Huynh, Mohar, Šámal, Thomassen, and Wood [93], who demonstrated that any countable graph G that contains every countable planar graph must also contain the infinite countable clique as a minor. In joint work with Louis Esperet, Carla Groenland, Claire Hilaire, Clément Rambaud and Alexandra Wesolek, we extend the study of "faithful" structures to the finite setting and establish the following finite analogue of the result from [93].

Theorem 5.4. *There is a polynomial function $f_{5.4}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let $t, \ell \geqslant 2$ be integers. If U is a K_t-minor-free graph containing every $\ell \times \ell$ triangulated grid as subgraph, then*

$$|V(U)| \geqslant 2^{\frac{1}{f_{5.4}(t)}\cdot \ell}.$$

In particular, for every integer $t \geqslant 5$ there exists a constant $C_t > 0$ such that for every integer $n \geqslant 2$, every K_t-minor-free graph containing every n-vertex planar graph as subgraph has at least $2^{C_t\sqrt{n}}$ vertices.

Together with Carla Groenland and Rajko Nenadov, we studied also "faithful" universal structures for the class of all posets on n elements. Specifically, we considered the following natural question: "What is the minimum size of a poset U that contains every poset on n elements as an induced subposet?" Unlike the case of planar graphs mentioned above, we cannot expect U to have polynomial size, as the number of distinct posets on n elements is of order $2^{(1/4+o(1))n^2}$, which implies a lower bound of $2^{(1/4+o(1))n}$ on the size of U. To our knowledge, the best known upper bound prior to our work was 2^n, achieved by the hypercube of dimension n.

Theorem 5.8. *For any $n \in \mathbb{N}$, there exists a poset U_n containing all n-element posets as an induced subposet, and $|U_n| \leqslant 2^{\frac{2}{3}n+O(\sqrt{n})}$.*

1.5 Result not included in this thesis

In addition to the projects presented in this manuscript, I have also worked on other topics in Combinatorics and Theoretical Computer Science. I have studied problems in distributed

[1] See Section 5.1 for additional references.

algorithms from a pure mathematics perspective, using simplicial topology to extend a meta-theorem in the field [23] and probability theory to investigate a self-stabilising clock synchronisation problem [24]. I have also explored other graph reconstruction models than the distance reconstruction model presented in this thesis [21] and applied structural graph theory to analyse graph parameters such as the burning number [20] and path eccentricity [29].

Chapter 2

Graph Reconstruction via distance oracle

This chapter focuses on graph reconstruction via distance oracle queries. The work presented is based mainly on published results, co-authored with Carla Groenland [25, 26] and a sole-author result [19].

How can we determine the structure of a decentralised network (such as the Internet or sensor networks) with minimal overhead? Such reconstruction problems have been extensively studied (e.g. [31, 64, 140, 141, 161, 25, 126]). The vertices of the network are distinct networks (autonomous systems) and the edges represent peering relations (voluntary interconnection). Tools such as `traceroute` (also called `tracert`) are used to record the route through the Internet from one network to another.

Due to privacy and security concerns, the full path information may not always be available and only delayed information may be given. A ping-pong protocol is one of the most basic tools in a peer-to-peer or Internet network. It is a two-node protocol where one node sends a dummy message to the second one. Once the message is received, the second node directly responds with a dummy message to the first node. The goal of this process is to compute the time between the departure of the first message and the arrival of the second one. From this, the first node can deduce an estimate of the distance between itself and the second node in the network. In this chapter, we are interested in the following question: how fast can you reconstruct a hidden network only using a ping-pong protocol? This question arises naturally not only from computer networks problems but also appears in phylogenetics [90, 182, 117] and in machine learning [170] where it has major applications.

2.1 Introduction

The distance query model In order to capture this kind of ping-pong protocol, the distance query model has been introduced [31]. In this framework, only the vertex set V of a hidden graph $G = (V, E)$ is known and the aim is to reconstruct the edge set E via distance queries to an oracle. For a pair of vertices $(u, v) \in V^2$, the oracle answers the shortest path distance between u and v in G and the algorithm can select the next query based on the responses of earlier queries. If there is a unique graph consistent with the query responses, the graph has been reconstructed. We denote by $\textsc{Query}_G(u, v)$ the query to our oracle that answers $d_G(u, v)$. We also slightly abuse notation and denote by $\textsc{Query}_G(A, B)$ the set of queries $\{\textsc{Query}_G(a, b) \mid (a, b) \in A \times B\}$.

For a graph class $\mathcal{G}$ of connected graphs, we say that an algorithm reconstructs the graphs in the class if for every graph $G \in \mathcal{G}$, the distance profile obtained from the queries uniquely identifies G within $\mathcal{G}$. The *query complexity* is the maximum number of queries that the algorithm takes on an input graph from $\mathcal{G}$. For a randomised algorithm, the query complexity is defined by the maximum over all inputs of the expected number of queries (with respect to the randomness in the algorithm).

The first observation about our oracle is that by querying the distance between every pair (u, v) of vertices in G, we can fully reconstruct the edge set as $E = \{\{u, v\} \mid d(u, v) = 1\}$.

Proposition 2.1 (Folklore). *The class of all n-vertex graphs can be reconstructed in $\binom{n}{2}$ queries.*

This implies a trivial upper bound of $O(|V|^2)$ on the query complexity. Unfortunately, this upper bound is tight in general. For example, the clique K_n is indistinguishable from K_n minus an edge $\{u, v\}$: every query answer is the same for both graphs, except for the pair (u, v). Thus, any algorithm must make at least $\Omega(|V|^2)$ queries to reconstruct these graphs. This trivial upper bound happens to be tight even on sparse graphs such as trees. For example, to distinguish between the two trees represented in Fig. 2.1 $\Theta(n^2)$ queries are needed).

Proposition 2.2 ([156]). *Any algorithm reconstructing the class of all n-vertex trees uses $\Omega(n^2)$ queries.*

Kannan, Mathieu, and Zhou [140, 108] showed that the core issue in both of the trivial upper bounds mentioned above is, in fact, the presence of high-degree vertices. To establish this result, they designed an efficient randomised algorithm capable of reconstructing all connected graphs of bounded degree and, for the first time, proved a non-trivial complexity upper bound for a large class of graphs. Note that the assumption of connectedness is also natural, as the empty graph on n vertices requires $\Omega(n^2)$ to be distinguished from the graphs containing one edge and $n - 2$ isolated vertices. For the rest of this chapter, unless stated otherwise, we always assume graphs to be connected.

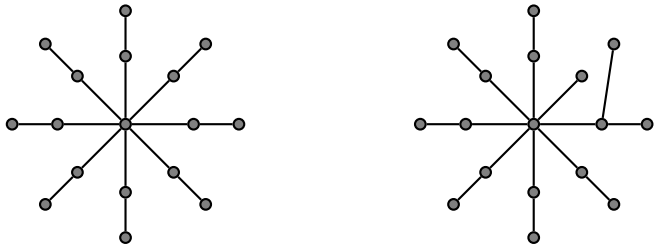


Figure 2.1: The two depicted trees are difficult to distinguish: only three pairs of distances differ.

Theorem 2.3 ([140, 108]). *There exists an algorithm reconstructing the class of connected n-vertex graphs of maximum degree Δ using $O(\Delta^4 n^{3/2} \operatorname{polylog} n)$ queries.*

Despite active research in the last decade, [161, 25, 26, 126] the algorithm developed by Kannan, Mathieu and Zhou remains the most efficient known solution to the problem of reconstructing bounded degree graphs.

We will now give an overview of the proof of Theorem 2.3. Its main insight lies in cleverly translating a result from Thorup and Zwick about compact routing [177] into a distance query algorithm. To understand this result, we need to define the following concept. The Voronoi cell of $s \in S \subseteq V(G)$ in a graph G is defined as:

$$\operatorname{Vor}_G(s, S) = \{u \in V(G) \mid d_G(u, s) = d_G(u, S)\}.$$

In [177], Thorup and Zwick proved that for any $r \in \mathbb{N}$ an algorithm could efficiently compute a set $\{s_1, \ldots, s_r\} \subseteq V(G)$ of *centers* such that the sizes of their Voronoi cells were roughly balanced. Mathematically $\{s_1, \ldots, s_r\}$ satisfy, for every $i \in [r]$, $|\operatorname{Vor}_G(s_i, S)| \leqslant \frac{n}{r}\Delta \log n$. Mathieu, Kannan and Zhou proved that Thorup's algorithm can be implemented efficiently using distance queries [140, 108].

Lemma 2.4 ([140, 108]). *There exists a randomised algorithm R that takes as input a hidden connected graph G on n-vertex and an integer $r \in [n]$. Let Δ denote the maximum degree of G then R outputs a set of centers $S := \{s_1, \ldots, s_r\}$ and the neighbourhood of their Voronoi cells $\{N[\operatorname{Vor}_G(s_i, S)] \mid i \in [r]\}$, satisfying*

$$\forall i \in [r], |\operatorname{Vor}_G(s_i, S)| \leqslant \frac{n}{r}\Delta \log n.$$

Moreover, R uses $O(\Delta r n \operatorname{polylog} n)$ queries w.h.p..

Theorem 2.3 is then easily derived from Lemma 2.4.

Proof of Theorem 2.3. First, execute Lemma 2.4 with $r = \lceil \sqrt{n} \rceil$, and recover $C_i = N[\mathrm{Vor}_G(s_i, S)]$ for all $i \in [r]$. Note that for every edge $uv \in E(G)$, there exists $i \in [r]$ such that $u, v \in C_i$. Therefore the naive algorithm which consists of, for all $i \in [\lceil \sqrt{n} \rceil]$, querying all pairs of elements in C_i (i.e. $\textsc{Query}(C_i, C_i)$), reconstructs G completely. Moreover,

$$|C_i| = |N[\mathrm{Vor}_G(s_i, S)]| \leqslant (\Delta + 1)|\mathrm{Vor}_G(s_i, S)| \leqslant 2\frac{n}{r}\Delta^2 \log n \leqslant 2\sqrt{n}\Delta^2 \log n.$$

The query complexity of the algorithm is

$$\lceil \sqrt{n} \rceil \left(2\sqrt{n}\Delta^2 \log n\right)^2 + \Delta n^{3/2} \operatorname{polylog} n = O(\Delta^4 n \operatorname{polylog} n),$$

where the first term of the sum comes from asking $\textsc{Query}(C_i, C_i)$ for all $i \in [\lceil \sqrt{n} \rceil]$ and the second one corresponds to the execution of Lemma 2.4. □

Around Theorem 2.3 To further improve upon the algorithm presented in Theorem 2.3, it would be natural to consider a more subtle approach than the brute-force method in order to reconstruct the edges in a given cell C_i. Certainly, a more sophisticated algorithm could outperform the naive approach of asking $\textsc{Query}(C_i, C_i)$ to reconstruct the edges within C_i. In this direction, Kannan, Mathieu and Zhou [108] conjectured the following:

Conjecture 2.5. *For any $\Delta \in \mathbb{N}$ there exists an algorithm A that reconstructs the class of all connected n-vertex graphs of maximum degree Δ using $O_\Delta(n \operatorname{polylog} n)$ queries w.h.p..*

Unfortunately, more than ten years later, Theorem 2.3 remains the best-known algorithm. One of the reasons for this is the difficulty of reconstructing each Voronoi cell independently. Surprisingly, it can be shown that if we restrict ourselves to querying only pairs of vertices within C_i, the trivial upper bound of $\Theta(|C_i|^2)$ queries is optimal up to a polylogarithmic factor.

In order to formalise the previous statement, let us introduce the concept of partial oracles.

Definition 2.6. *Given a graph G and a subset of the vertices $X \subseteq V(G)$ the partial oracle $\textsc{Query}_{G|X}$ is the restriction of $\textsc{Query}_G$ to the set $X \times X$.*

The following theorem exhibits a class of graphs, and a given Voronoi cell $X \subseteq V(G)$, such that reconstructing X from $\textsc{Query}_{G|X}$ requires $\Theta(|X|^2/\log^2 |X|)$, this is a joint work with Carla Groenland.

Theorem 2.7. *For any $n \in \mathbb{N}$, there exists a family of graphs $\mathcal{G}$, all on the same vertex set $V := X \sqcup Y$, and there exists $x \in V$ and $S \subseteq V$ satisfying the following properties, for any $G, G' \in \mathcal{G}$:*

- *G has maximum degree 5,*
- *$X = \mathrm{Vor}_G(x, S)$,*
- *$G[Y] = G'[Y]$,*
- *$|V| = \Theta(n^2)$ and $|X| = \Theta(n \log n)$.*

Moreover, any randomised reconstruction algorithm using the partial oracle $\textsc{Query}_{G|X}$, taking as input a graph $G \in \mathcal{G}$, uses $\Omega(n^2)$ queries to uniquely reconstruct $G[X]$ in expectation.

Reconstructing k-chordal graphs Theorem 2.7 tells us that new insights are necessary to break through the $n^{3/2+o(1)})$ upper bound. In an attempt to develop new techniques for reconstructing graphs from distance queries, researchers have sought to design efficient algorithms that match the query complexity of $O(n \operatorname{polylog} n)$ proposed in Conjecture 2.5 for subclasses of graphs with bounded degree [108, 141, 161]. For example, Kannan, Mathieu, and Zhou [108] designed a randomised algorithm for reconstructing the class of n-vertex chordal graphs with maximum degree Δ, using $O(4^{\Delta}\Delta^3 n \log^2 n)$ queries with high probability. This randomised complexity was later improved by Rong, Li, Yang, and Wang to $O(\Delta^2 n \log^2 n)$ queries with high probability [161]. We present the first deterministic algorithm for reconstructing graph classes. Applying our technique to chordal graphs with maximum degree Δ leads to the following result.

Theorem 2.8. *There exists a deterministic algorithm to reconstruct the class of connected n-vertex chordal graphs of maximum degree at most $\Delta \leqslant \sqrt{\log n}$ using $O(\Delta n \log n)$ distance queries.*

Our technique allows us to design efficient algorithms for a more general class of graphs, namely k-chordal graphs, graphs that exclude C_r as an induced subgraph for every $r \geqslant k$.

Theorem 2.9. *There exists a deterministic algorithm to reconstruct the class of connected k-chordal graphs of maximum degree at most Δ on n vertices using $O_{\Delta,k}(n \log n)$ queries.*

In the more restricted class of bounded degree trees we obtain the following result.

Theorem 2.10. *There exists a deterministic algorithm to reconstruct the class of trees of maximum degree at most Δ on n vertices using $\Delta n \log_{\Delta} n + (\Delta + 2)n$ queries.*

We would like to mention here that, in the light of a lower bound result mentioned later (see Theorem 2.14), the three results presented above have an optimal dependence on n, even among randomised algorithms. Moreover, Theorem 2.10 also has an optimal dependence on Δ, and Theorem 2.8 is optimal up to a $O(\log \Delta)$-factor.

By combining the intuition from the previous result with the techniques developed in [108], we are able to prove the following result. The *treelength* of a graph is defined formally in Section 2.2, intuitively, G has treelength k if every subgraph of G admits a $\frac{1}{2}$ balanced separator that is local (i.e. the maximum distance in G between two vertices of the separator is k). In particular, k-chordal graphs have treelength at most k.

Theorem 2.11. *For any $\Delta, n, k \geqslant 0$, there exists a randomised algorithm that reconstructs the class of connected n-vertex graphs of maximum degree at most Δ and treelength at most k using $O_{\Delta,k}(n \log^2 n)$ distance queries in expectation.*

The proof of Theorem 2.11 is omitted from this manuscript, but can be find in [26].

Reconstructing random graphs Mathieu and Zhou [141] initiated the study of the complexity of distance reconstruction on random graphs [126, 19, 172] by showing that random Δ-regular graphs (where Δ is a constant) can be reconstructed using only $O(n \log^2 n)$ queries in expectation. The algorithm they used is simple and natural, but the complexity analysis requires a deep understanding of the structural properties of random Δ-regular graphs. In [141], the authors mentioned that their algorithm could potentially lead to $O(n \operatorname{polylog} n)$ query upper bounds in different randomised settings, including $G(n,p)$ for values of p close to the connectivity threshold.

Recently, Krivelevich and Zhukovskii [126] studied the query complexity of reconstructing $G \sim G(n,p)$ for large values of $p \geqslant n^{-1+\varepsilon}$ and derived tight bounds for p outside of some threshold points around which the diameter increases, explicitly: $p = n^{-\frac{k}{k+1}+o(1)}$ for $k \in \mathbb{N} \cup \{\infty\}$. In this range, they managed to pinpoint precisely the complexity to be $\Theta(n^{4-d}p^{2-d})$ queries with high probability, where d is the diameter of G. For these values of p, the diameter is known to be a fixed constant independent of n w.h.p.. They also studied this problem in the case of a non-adaptive algorithm,[1] and proved a bound of $\Theta(n^{4-d}p^{2-d}\log n)$ queries outside of the diameter increase threshold points mentionned above. The authors asked if their result could be further extended to values of p outside of these threshold windows. In particular, the range around the connectivity threshold $p = \Theta\left(\frac{\log n}{n}\right)$ is outside of the scope of their proof. Independently from Krivelevich and Zhukovskii [126], we studied the complexity of reconstructing $G \sim G(n,p)$, and proved the following theorem.

Theorem 2.12. *For any $\varepsilon \geqslant 0$, and every $n \in \mathbb{N}$, for every $\frac{2000 \log n}{n} \leqslant p \leqslant n^{-\frac{1}{2}-\varepsilon}$, there exists an algorithm that reconstructs $G \sim G(n,p)$ using $O(\Delta^2 n \log n)$ queries in expectation, where $\Delta = (n-1)p$ is the expected average degree of G.*

[1] an algorithm where queries can be seen as simultaneous and do not depend on each other's answers.

Theorem 2.12 covers a large continuous range of values of p. Our result compares to the result proved by Krivelevich and Zhukovskii [126]. When p is close to the thresholds $p = n^{-\frac{k}{k+1}+o(1)}$ we obtain the same complexity, up to a $n^{o(1)}$ factor. However, Krivelevich and Zhukovskii proved tighter bounds when p is away from this threshold and k is a constant. In contrast to [126], Theorem 2.12 is designed to address very small values of p, particularly those close to the connectivity threshold (around $\Theta(\log n/n)$). In this regime, its query complexity is only a factor $O(\Delta/\log\log n)$ away from the lower bound of $\Omega(\Delta n \log n/\log\log n)$, which can be derived by a result of Kannan, Mathieu, and Zhou [108]. Therefore Theorem 2.12 applied to $p \in [\frac{2000\log n}{n}, \frac{\log^2 n}{n}]$ partially answers the question of Krivelevich and Zhukovskii [126] regarding extending their bounds outside the range $p \gg \log^2 n/n$. It also confirms the intuition of Mathieu and Zhou, mentioned in [141] that their algorithm could potentially be applied to this range of p.

Corollary 2.13. *For every $n \in \mathbb{N}$, for $p \in [\frac{2000\log n}{n}, \frac{\log^2 n}{n}]$ there exists an algorithm that reconstructs $G \sim G(n,p)$ using $O(n\log^5 n)$ queries in expectation.*

Lower bound Very few results are known about query complexity lower bounds. Prior to the work we present now, the best lower bound for reconstructing bounded degree graphs was proved in [107] using an information-theoretic argument. They showed that $\Omega_\Delta(n\log n/\log\log n)$ queries are needed to reconstruct n-vertex trees of maximum degree Δ. Let us consider $\Delta = 10$ to sketch the idea of the proof. We use that the class $\mathcal{C}$ of n-vertex graphs of maximum degree 10 and diameter at most $\log n$ has size $\Omega(2^{n\log n})$. For any algorithm distinguishing graphs from $\mathcal{C}$ in N queries, no two distinct graphs from $\mathcal{C}$ can get the same responses to the first N queries. The diameter condition ensures that (for graphs in $\mathcal{C}$) every query answer is an element of $\{0,\dots,\lfloor\log n\rfloor\}$ and therefore can be encoded using at most $\log(\log n+1)$ bits. Concatenating the answers to the first N queries, and using that the number of possible strings needs to be at least the number of graphs in the class, we find $2^{N\log(\log n+1)} = \Omega(2^{n\log n})$. This implies $N = \Omega(n\log n/\log\log n)$.

Improving on such an information-theoretic lower bound is often difficult. More generally, randomised query complexity is infamously difficult to pinpoint: for example, state-of-the-art results are also far from tight bounds for the recursive majority function [85, 129, 134]. In the setting of the evasiveness conjecture, the oracle can answer *adjacency queries* ("given u, v, is $\{u,v\} \in E$?") instead of distance queries. It has been shown already in 1975 [157] that for any fixed non-trivial monotone graph property of the graph (such as "Does G have a triangle") any deterministic algorithm needs $\Theta(n^2)$ on n-vertex graphs. At the same time, the best randomised query lower bound $\Omega(n^{4/3}(\log n)^{1/3})$ from [46] is far from the best upper bound of $O(n^2)$. Even seemingly simple questions such as estimating the average degree of a graph using vertex degree queries requires new probabilistic tools to

achieve tight bounds [72, 84].

For n-vertex trees of maximum degree Δ, we achieve the correct dependency on n and Δ for both the randomised and deterministic query complexity using distance queries.

Theorem 2.14. *Let $\Delta \geqslant 2$ and $n = 2c\Delta^k$ be integers, where $c \in [1, \Delta)$ and $k \geqslant 50(c \ln c + 3)$ is an integer. Then, any randomised algorithm requires at least $\frac{1}{50}\Delta n \log_\Delta n$ queries in expectation to reconstruct n-vertex trees of maximum degree $\Delta + 1$.*

We give more details about this lower bound in Section 2.5, where we also prove a more general statement that allows us to derive bounds not only for the distance reconstruction model but also for many other models arising from phylogenetic reconstruction and beyond.

Note that, in Theorem 2.14, for any $n \geqslant 2$, there is a unique $(c, k) \in [1, \Delta) \times \mathbb{Z}_{\geqslant 0}$ with $n/2 = c\Delta^k$ thus the only assumption in our lower bound is that n is sufficiently large compared to Δ. Our result allows Δ to grow slowly with n (e.g. $\Delta = O((\log n)^\alpha)$ with $\alpha \in (0, 1)$). Moreover, we allow Δ to be larger for specific values of n (e.g. $O(n^{1/150})$ for $c = 1$). We made no attempt to optimise the constant.

As mentioned before, Theorem 2.14 ensures that Theorem 2.10 is optimal up to a $O(1)$ factor. It also ensures that Theorem 2.8 is optimal up to a factor $O(\log \Delta)$ and Theorem 2.9 is optimal up to a factor $O_{\Delta,k}(1)$. In particular, the asymptotic dependence on n is tight for the three theorems.

Roadmap In Section 2.2 we set up our notations formally and give the relevant definitions and tools used throughout the chapter. In Section 2.3, we present our method to design new deterministic algorithms, proving Theorem 2.9 and Theorem 2.10, the proof of Theorem 2.8 is omitted from this manuscript but can be found in [25]. In Section 2.4 we discuss our upper bound on the query complexity in the binomial random graph setting, proving Theorem 2.12. In Section 2.5, we prove the randomised lower bound stated in Theorem 2.14 and discuss the implication of this bound in the distance reconstruction model and beyond.

2.2 Preliminaries

This section aims to introduce the notation and terminology used in this chapter. We will design and analyse algorithms from a structural and a probabilistic perspective. Therefore, these preliminaries are divided into three parts. The first part introduces our notations for the distance oracle. The second part defines the class of graphs which is the main focus of our algorithms, while the third part discusses the probabilistic notions useful to this chapter.

Query complexity In reconstruction problems, an unknown combinatorial object can only be accessed through a restricted oracle. The goal is to determine properties of the object or even identify it completely using an efficient algorithm. Unlike traditional algorithms, where efficiency is typically measured by the number of basic instructions executed, reconstruction algorithms assume that computation is cheap while queries to the oracle are expensive, aligning with most real-world applications. Consequently, instead of computational cost, we focus on the *query complexity*, which is defined as the number of queries made to the oracle. Similar to traditional complexity measures, we define the query complexity of a deterministic algorithm as the maximum number of queries it makes over all possible hidden graphs G.

Distance reconstruction We recall the notation we use for our oracle in this chapter. We denote by $\textsc{Query}_G(u, v)$ the query to the oracle that answers the shortest distance between the vertices u and v in G. We also slightly abuse notation and denote by $\textsc{Query}_G(A, B)$ the set of queries $\{\textsc{Query}_G(a, b) \mid (a, b) \in A \times B\}$ and for $F \subseteq \binom{V(G)}{2}$, $\textsc{Query}(F)$ denote the set of queries $\{\textsc{Query}(u, v) | \{u, v\} \in F\}$. We denote by d_G the distance function of the graph G.

Tree decompositions and chordal graphs A *tree decomposition* of a graph G is a tuple $(T, (B_t)_{t\in V(T)})$ where T is a tree and B_t is a subset of $V(G)$ for every $t \in V(T)$, for which the following conditions hold.

- For every $v \in V(G)$, the set of $t \in V(T)$ such that $v \in B_t$, is non-empty and induces a subtree of T.
- For every $uv \in E(G)$, there exists a $t \in V(T)$ such that $\{u, v\} \subseteq B_t$.

This notion was introduced by [158], and we say that a tree decomposition is trivial if T has a unique vertex.

The *treelength* of a graph G (denoted $\mathrm{tl}(G)$) is the minimal integer k for which there exists a tree decomposition $(T, (B_t)_{t\in V(T)})$ of G such that $d_G(u, v) \leqslant k$ for every pair of vertices u, v that share a bag (i.e. $u, v \in B_t$ for some $t \in V(T)$). We refer the reader to [65] for a detailed overview of the class of bounded treelength graphs.

Tree decompositions are linked with the notion of *balanced separator*. For $\beta \in (0, 1)$, a *β-balanced separator* of a graph $G = (V, E)$ for a vertex set $A \subseteq V$ is a set S of vertices such that the connected components of $G[A \setminus S]$ are of size at most $\beta|A|$.

Theorem 2.15. *[158] For any graph G and any non trivial tree decomposition $(T, (B_t)_{t\in V(T)})$ of G, there exist and edge $e := tt' \in E(T)$ such that, $B_t \cap B_{t'}$ is a $\frac{1}{2}$-balanced separator of G.*

In this chapter, we will be studying chordal graphs. This class of graphs has multiple equivalent definitions. One definition states that a graph G is chordal if it is maximal for some tree decomposition $(T, (B_t)_{t\in V(T)})$ (i.e. all edges authorised by $(T, (B_t)_{t\in V(T)})$ are present: for all $t \in T$, $\forall u, v \in B_t$ with $u \neq v$ we have $uv \in E(G)$). Alternatively, chordal graphs can be more simply defined as the class of graphs that excludes C_4,...,C_k for any $k \geqslant 3$ as an induced subgraph. This second definition admits a natural generalisation.

Definition 2.16. *For any $k \in \mathbb{N}$, a graph G is said to be k-chordal if it excludes C_ℓ as an induced subgraph for every $\ell \geqslant k$.*

The class of k-chordal graphs has been extensively studied from a structural point of view. In particular, the following structural result will prove useful throughout the chapter.

Lemma 2.17 ([123])**.** *For any $k \in \mathbb{N}$, any k-chordal graph has treelength at most k.*

Random graphs For $n \in \mathbb{N}$ and $p \in [0, 1]$, we denote by $\mathrm{Bin}(n, p)$ the binomial distribution of parameter n and p and by $\mathrm{B}(p)$ the Bernoulli distribution of parameter p. We denote by $G(n, p)$ the Erdős-Rényi model, that is, a random distribution over graphs on the vertex set $[n]$ where each edge is present independently of each other with probability p. The following results are standard, and the reader is invited to learn more about random graphs in [37].

Lemma 2.18. *For every $n \in \mathbb{N}$, for $p \geqslant \frac{2000 \log n}{n}$, let $G \sim G(n, p)$. Then, G has max degree at most $\frac{3}{2}p(n-1)$ and min degree at least $\frac{1}{2}p(n-1)$ with probability $1 - o(n^{-1})$.*

Proof. Fix a vertex $u \in V(G)$, the degree of u is the sum of the independent $\{0, 1\}$ variables $(X_v)_{v\in V(G)\setminus\{u\}}$ defined by $X_v = 1$ if and only if $uv \in E(G)$. Note that $X_v \sim B(p)$ for all v. We apply Chernoff's bound with $\delta = \frac{1}{2}$ and $\mu = p(n-1)$.

$$\mathbb{P}\left(\left|\sum_{v\in V(G)\setminus\{u\}} X_v - p(n-1)\right| \geqslant \frac{1}{2}p(n-1)\right) \leqslant 2e^{-p(n-1)/12} = o(n^{-2}),$$

where the last inequality comes from $p \geqslant 2000 \log n/n$. It now suffices to take a union bound over all vertices to conclude. □

Lemma 2.19. *For $n \in \mathbb{N}$, for any $\Delta > 0$, any $p \leqslant \frac{1}{n\Delta}$ and $G \sim G(n, p)$, the number of non-isolated vertices in G is at most $4n/\Delta$ with probability at least $1 - 2e^{-n/(3\Delta)}$.*

Proof. Let X_p denote the random variable counting the number of isolated vertices in $G(n, p)$. Note that $X_{\frac{1}{n\Delta}}$ stochastically dominates X_p for $p \leqslant \frac{1}{n\Delta}$ (i.e for any $x \in \mathbb{R}$, $\mathbb{P}[X_{\frac{1}{n\Delta}} \geqslant x] \geqslant \mathbb{P}[X_p \geqslant x]$). Therefore we will only focus on $p = \frac{1}{n\Delta}$ for the sake of this proof.

Note that an edge in G creates at most two non-isolated vertices, therefore the number of non-isolated is at most twice the number of edges of G. Moreover we have $\mathbb{E}[|E(G)|] = p\binom{n}{2} \leqslant n^2$. We bound the number of edges of G using Chernoff's bound (Lemma 0.1) with $\delta = 1$ and $\mu = p\binom{n}{2}$.

$$\mathbb{P}\left(||E(G)| - \mu| \geqslant \frac{n}{\Delta}\right) \leqslant e^{-\frac{n}{3\Delta}}$$

which, using the remark above, directly implies the lemma statement. □

2.3 From reconstructing trees to reconstructing k-chordal graphs

This section presents deterministic algorithms for reconstructing classes of graphs. We will first present, in Section 2.3.1, an algorithm that reconstruct the class of bounded degree trees, as it encapsulates most of the algorithmic ideas developed in this section, while being fairly simple. In Section 2.3.2 we generalise the techniques used for trees to a much larger class of graphs, namely k-chordal graphs.

In this section $N_G^k(u)$ (resp. $N_G^{\leqslant k}(u)$) denote the set of vertices at distance exactly k (resp. at most k) from $u \in V(G)$. We omit the subscript G when G is clear from context.

2.3.1 Distance reconstruction for trees

Theorem 2.10 ([25]). *There exists a deterministic algorithm to reconstruct the class of trees of maximum degree at most Δ on n vertices using $\Delta n \log_\Delta n + (\Delta + 2)n$ queries.*

Proof. Let T be a tree on n vertices, and let Δ be the maximum degree of T. Our algorithm starts as follows. We pick an arbitrary vertex $v_0 \in V(T)$ and will consider (for the analysis) the input tree T as rooted in v_0. We call $\textsc{Query}(v_0, V(T))$. We define the i^{th} layer of T as $L_i = \{v \in V(T) \mid d(v_0, v) = i\}$. We proceed to reconstruct the graph induced by the first i layers by induction on i.

Note that $T[L_0] = (\{v_0\}, \emptyset)$ is immediately reconstructed. We fix an integer $i \geqslant 1$ and assume that the first $i-1$ layers are fully reconstructed (i.e we discovered all the edges and non-edges of $T[L_0 \cup \cdots \cup L_{i-1}]$). Let $T' = T[L_0 \cup \cdots \cup L_{i-1}]$ be the already reconstructed subtree. We show how to reconstruct the edges between the $(i-1)^{\text{th}}$ layer and the i^{th} layer. Note that this suffices to reconstructs all the edges (since in a tree, edges can only be between consecutive layers).

Choose an arbitrary vertex $v \in L_i$. We first show that we can find the parent of v in L_{i-1} using $O(\Delta \log n)$ queries and then describe how to shave off an additional $(\log \Delta)$-factor.

The procedure goes as follows. Let $T_1 = T'$. As T is a tree, it admits a $\frac{1}{2}$-balanced separator of size 1. Let s_1 be a vertex for which $\{s_1\}$ forms such a separator. We ask first $\textsc{Query}(v, N[s_1])$, where the neighbourhood is taken in T_1. As T is a tree, there is a unique path between any two vertices. So for $w \in N(s_1)$, the distance $d(v, w) = d(v, s_1) - 1$ if w lies on the shortest path from v to s_1 and $d(v, s_1) + 1$ otherwise. From this, we can infer the neighbour x of s_1 that is the closest to v as the one for which the answer is smallest (or find that s_1 is adjacent to v and finish). Moreover, the unique path from s_1 to v lives in the connected component T_2 of $T_1 \setminus \{s_1\}$ that contains x. In particular, T_2 contains the parent of v (see Fig. 2.2).

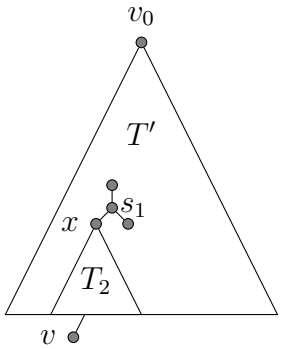


Figure 2.2: The subtree T_2 contains the neighbour of v on a shortest path to s_1 and so contains the parent of v.

We can repeat this process and construct two sequences $(T_j)_{j\in\mathbb{N}}$ and $(s_j)_{j\in\mathbb{N}}$, where T_{j+1} is the connected component of $T_j \setminus \{s_j\}$ containing the parent of v and $s_j \in V(T_j)$ is chosen so that $\{s_j\}$ is a $\frac{1}{2}$-balanced separator of T_j. Once T_ℓ contains less than $\Delta + 1$ vertices for some ℓ or the vertex s_ℓ is identified as the parent of v, we finish the process[2]. By definition of $\frac{1}{2}$-balanced separator,

$$\forall j \in [\ell - 1], |T_{j+1}| \leqslant |T_j|/2 \quad \text{and thus} \quad \ell \leqslant \log n.$$

If the process finished because T_ℓ has at most Δ vertices, we use at most Δ additional queries via $\textsc{Query}(T_\ell, v)$. We infer the parent of v from the result. For each $j \leqslant \ell$, we use at most $\Delta + 1$ queries to reconstruct T_{j+1} from T_j. Hence we use $O(\Delta \log n)$ queries in total.

Taking a closer look at the process, at any step j, we can choose the order on the queries $\textsc{Query}(v, w)$ for $w \in N(s_j)$ and may not need to perform all the queries. Given a subtree S of T' on $s \geqslant 1$ vertices that contains the parent of v, we now show how to find the parent of v

[2]If desired, we may define $T_j = T_\ell$ and $s_j = s_\ell$ for all $j \geqslant \ell$.

in $f(s) = \Delta \log_\Delta s + \Delta + 1$ queries (giving the desired improvement of a $(\log \Delta)$-factor). If S has at most $\Delta + 1$ vertices, we may simply $\textsc{Query}(v, S)$ and deduce the answer. Otherwise, let $s \in S$ be a $\frac{1}{2}$-balanced separator for S. This has at least two neighbours since S has at least $\Delta + 1$ vertices. We order the connected components of $S \setminus \{s\}$ by decreasing size, and ask the queries in the same order: we start with $\textsc{Query}(v, w_1)$ for w_1 the neighbour of s which is in the largest component, then proceed to the neighbour of the second largest component etcetera. We terminate when we find two different distances or queried all the neighbours. In particular, we never perform $\textsc{Query}(v, s)$.

- If $d(v, w)$ for $w \in N(s)$ are all the same then s is the parent of v. We terminate and recognise s as the parent of v. We used at most $\Delta \leqslant f(s)$ queries.
- If we discover that $d(v, w) < d(v, w')$ for some $w, w' \in N(s)$, then s is not the parent of v. In fact, w is the vertex from $N[s]$ closest to v and we recursively perform the same procedure to the subtree S' of $S \setminus \{s\}$ that contains w. Note that S' must contain the parent of v.

If we query 2 neighbours of s before detecting the component containing the parent of v, our next subtree S' satisfies $|S'| \leqslant |S|/2$ since $\{s\}$ is a balanced separator. If we query $m \geqslant 3$ neighbours of s before detecting the component containing the parent of v, our next subtree S' satisfies $|S'| \leqslant |S|/m$ since there are $m - 1$ components of $S \setminus \{s\}$ that are at least as large. Either way, we decrease the size of the tree by a factor at least x if we perform x queries, where $x \in \{2, \ldots, \Delta\}$.

We show by induction on s that the procedure described above uses at most $f(s) = \Delta \log_\Delta s + \Delta + 1$ queries. The claim is true when $s \leqslant \Delta + 1$. By the discussion above, for $s \geqslant \Delta + 2$, the process either finishes in Δ queries or uses $x + f(s')$ queries for some $s' \leqslant s/x$ and $x \in \{2, \ldots, \Delta\}$. It thus suffices to show that

$$f(s/x) + x \leqslant f(s) \text{ for all } x \in \{2, \ldots, \Delta\}.$$

By definition, $f(s) - f(s/x) = \Delta \log_\Delta x$. We show that $\Delta \log_\Delta x \geqslant x$ for all $x \in [2, \Delta]$. By analysing the derivative of $\Delta \log_\Delta x - x$ on the (real) interval $x \in [2, \Delta]$, we find that the minimum is achieved at $x = \Delta$, as desired.

With the improved procedure, we can reconstruct the edge from L_{j-1} to v in at most $\Delta \log_\Delta n + \Delta + 1$ queries. Repeating the same strategy to reconstruct the parent of every vertex, we obtain the edge set of T in at most

$$(n-1) + (n-1)(\Delta \log_\Delta n + (\Delta + 1)) \leqslant \Delta n \log_\Delta n + (\Delta + 2)n$$

queries. □

Note that even though we will show in Section 2.5 that we cannot achieve a better dependency in (n, Δ) using randomisation, we can improve the average query complexity by almost a factor of 2.

Theorem 2.20. *For any $\Delta \geqslant 4$, there exists a randomised algorithm for reconstructing n-vertex trees of maximum degree at most Δ using $\left(\frac{1}{2}\Delta/\log(\Delta)\right) n \log_2 n + (\Delta + 2 + \log_2 n)n$ queries in expectation.*

Proof. The algorithm works similarly to the algorithm from Theorem 2.10. We define the same layers and inductively reconstruct the graph induced on the first i layers. We find the parent of a vertex $v \in L_i$ via a similar sequence of separators $s_1, \dots, s_j$ and trees $T_1 \supseteq T_2 \supseteq \cdots \supseteq T_j$. The key difference is that when we wish to learn the vertex in $N[s_j]$ closest to v, then we perform $\textsc{Query}(v, w)$ for $w \in N[s_j]$ in an order that is chosen at random. Suppose that $|T_j| = b$.

Claim 2.21. *Let T be a tree and let $t \in V(T)$. Let $a_1, \dots, a_k$ be the sizes of the components of $T \setminus \{t\}$ and let $v_1, \dots, v_k$ denote the neighbours of s_j in these components. There is a random order on $v_1, \dots, v_k$ such that the expected number of vertices placed before v_i is at most $\frac{1}{2}\frac{1}{a_i}\sum_{j=1}^{k} a_j$ for all $i \in [k]$.*

Proof. We generate the order by independently sampling $X_i \sim U[0, a_i]$ uniformly at random for all $i \in [k]$, where $[0, a_i]$ denotes the set of real numbers between 0 and a_i. Almost surely, $X_{\pi(1)} > \cdots > X_{\pi(k)}$ for some permutation π on the support $[k]$ and this gives us our desired random order.

We prove that the expected number of vertices placed before v_1 is at most $\frac{1}{2}\frac{a_2+\cdots+a_k}{a_1}$ and then the remaining cases will follow by symmetry. Let $I(x_1)$ denote the number of vertices placed before v_1 given that $X_1 = x_1$, i.e. the number of $i \in \{2, \dots, k\}$ such that $X_i > x_1$:

$$I(x_1) = \sum_{i=2}^{k} \text{Bern}\left(\max\left(\frac{a_i - x_1}{a_i}, 0\right)\right).$$

A Bernoulli random variable with probability p has expectation p. The expected number of vertices placed before v_1 is hence

$$\frac{1}{a_1}\int_0^{a_1} \mathbb{E}[I(x_1)]dx_1 = \sum_{i=2}^{k} \frac{1}{a_1}\int_0^{\min(a_1,a_i)} 1 - \frac{x_1}{a_i}dx_1.$$

We now show for all $i \in [2, k]$ that the ith summand is at most $\frac{1}{2}\frac{a_i}{a_1}$, which implies that the number of vertices placed before v_1 is indeed at most $\frac{1}{2}\frac{a_2+\cdots+a_k}{a_1}$. We compute

$$\frac{1}{a_1}\int_0^{\min(a_1,a_i)} 1 - \frac{x_1}{a_i}dx_1 = \frac{\min(a_1, a_i)}{a_1}\left(1 - \frac{\min(a_1, a_i)}{2a_i}\right).$$

When $a_i \leqslant a_1$, the expression simplifies to $\frac{a_i}{a_1}\frac{1}{2}$ as desired. When $a_i \geqslant a_1$, the expression simplifies to $1 - \frac{1}{2}\frac{a_1}{a_i}$ which is at most $\frac{1}{2}\frac{a_i}{a_1}$ since

$$a_1 a_i \leqslant \frac{1}{2}a_i^2 + \frac{1}{2}a_1^2. \qquad \square$$

By the claim, if the parent of v is in a component T_{j+1} of $T_j - s_j$ of size a, then we query at most $\frac{1}{2}(b-a)/a$ vertices in expectation before we query the neighbour of s_j in T_{j+1}. This means that for a size reduction of $x = b/a$, we perform approximately $\frac{1}{2}x$ queries in expectation, compared to x in our deterministic algorithm. Using linearity of expectation, we will repeat a similar calculation to the one done in the proof of Theorem 2.10.

We show that we reconstruct the edge to the parent of v using at most $\frac{1}{2}\Delta \log_\Delta n + \Delta + 1 + \log_2 n$ queries in expectation. We show that once we have identified a subtree T_j containing the parent of v with $|T_j| = b$, we use at most $\Delta \log_\Delta b + \Delta + 1 + \log_2 b$ queries in expectation to find the parent of v. Let s_j be a $1/2$-balanced separator of T_j. The first base case is given when s_j is the parent of v: in this case we perform at most Δ queries (all neighbours of s_j) and find that s_j is the parent. The second base case is when $|T_j| = \Delta + 1$, in which case we query all distances between v and T_j and identify the parent of v.

We now assume that $b \geqslant \Delta + 2$ and that the parent of v is in a component T_{j+1} of $T_j \setminus s_j$. Let $a = |T_{j+1}|$. By the claim, we query at most $\frac{1}{2}(b-a)/a$ vertices in expectation before we query the neighbour w of s_j in T_{j+1}. We still need to query the distance from v to w. If w is the first vertex to be queried, then we need to query one more vertex. Since s_j is a $1/2$-balanced separator, this happens with probability at most $1/2$. So after at most $\frac{1}{2}\frac{b}{a} + 1$ in expectation we find the neighbour of v is in T_{j+1} after which by induction we need another $\frac{1}{2}\Delta \log_\Delta a + \Delta + 1 + \log_2 a$ queries in expectation. In total we use at most

$$\frac{1}{2}\frac{b}{a} + 1 + \frac{1}{2}\Delta \log_\Delta a + \Delta + 1 + \log_2 a$$

in expectation. We find that $\frac{b}{a} + \Delta \log_\Delta a \leqslant \Delta \log_\Delta b$ for all $\Delta \geqslant 4$ (same calculation as before) and $\log_2 a + 1 \leqslant \log_2 b$ since $a \leqslant b/2$. So we used at most $\Delta \log_\Delta b + \Delta + 1 + \log_2 b$ queries in expectation, as claimed. $\square$

2.3.2 Distance reconstruction for k-chordal graphs

In this subsection we extend the algorithm presented above in Theorem 2.10 from trees to k-chordal graphs: graphs without induced cycles of length at least $k+1$. In the simpler case of (3-)chordal graphs, (randomised) reconstruction from a quasi-linear number of queries was already known to be possible [108]. Besides extending the class of graphs, our algorithm

shaves off a $(\log n)$-factor and is now optimal in n (the number of vertices of the input graph).

The core of the proof uses the same principles as for trees in Section 2.3.1: we reconstruct the edges of a vertex u to the previous layer, layer-by-layer and vertex-by-vertex. The two important ingredients are (1) a structural result on the neighbourhood of a vertex (see Claim 2.22) and (2) the existence of "nice" balanced separators on the already reconstructed subgraph (see Claim 2.23). After removing the separator, we need to show that we can correctly determine the component that contains the neighbourhood of the vertex u that we are currently considering. We also need to reconstruct the edges within the layer, but this turns out to be relatively easy.

Theorem 2.9 ([25]). *There exists a deterministic algorithm to reconstruct the class of connected k-chordal graphs of maximum degree at most Δ on n vertices using $O_{\Delta,k}(n \log n)$ queries.*

Proof. We start by fixing a vertex v_0 and asking $\textsc{Query}(V(G), v_0)$. From that, we reconstruct $L_i = \{v \in V(G) \mid d(v, v_0) = i\}$. We write $L_{\bowtie i} = \cup_{j \bowtie i} L_j$ for any relation $\bowtie \in \{\leqslant, <, >, \geqslant\}$.

The algorithm proceeds by iteratively reconstructing $G[L_{\leqslant i}]$ for increasing values of i. Note that we can reconstruct $L_{\leqslant 2\Delta k}$, the vertices at distance at most $2\Delta k$ from v_0, using $O_{k,\Delta}(n)$ queries.

Suppose that we reconstructed $G_1 := G[L_{\leqslant i-1}]$ for some $i \geqslant 2\Delta k$ and we again want to reconstruct the two edge sets

$$E_{i-1,i} = \{uv \in E \mid u \in L_{i-1},\, v \in L_i\}$$

and

$$E_{i,i} = \{uv \in E \mid u, v \in L_i\}.$$

We call $H_1 = G[L_{\leqslant i-1-k}]$ the core of G_1. We need a lemma that implies that neighbourhoods are not spread out too much in G_1.

Claim 2.22. *For all $u \in L_i$ and $v, w \in N(u) \cap L_{i-1}$, $d_{G_1}(v, w) \leqslant \Delta k$.*

Proof. Let $v, w \in N(u) \cap L_{i-1}$ and let P be a shortest vw-path in G_1. If $V(P) \cap N(u) = \{v, w\}$, then the vertex set $V(P) \cup \{u\}$ induces a cycle in G, and so $|V(P)| \leqslant k$ (else the k-chordality would be contradicted). For the same reason, P can have at most $k-1$ consecutive vertices outside of $N(u)$. Since u has at most Δ neighbours, it follows that $d_{G_1}(v, w) \leqslant \Delta k$. □

Since G has treelength at most k, it has a tree decomposition $(T', \mathcal{B}')$ such that all bags $B' \in \mathcal{B}'$ satisfy $d_G(u, v) \leqslant k$ for all $u, v \in B'$. In particular the bags have size at most $\Delta^k + 1$.

We have already reconstructed G_1, so in particular we know $N^{\leqslant k}[v]$ for all $v \in H_1$. Therefore, we can construct a tree decomposition of $(T, \mathcal{B})$ of G_1 such that each $B \in \mathcal{B}$ has

size at most $\Delta^k + 1$ and for any bag $B \in \mathcal{B}$ that contains at least one vertex of the core H_1, we have $d_{G_1}(u, v) \leqslant k$ for all $u, v \in B$.

Fix $u \in L_i$. We describe an algorithm to reconstruct $N(u) \cap L_{i-1}$. The algorithm recursively constructs a sequence of connected graphs $(G_j)_{j=0}^{\ell}$ and a sequence of separators $(S_j)_{j=1}^{\ell}$ for some $\ell \leqslant \lceil \log(n) \rceil$, such that S_j is a $\frac{1}{2}$-balanced separator of G_j, $S_j \subseteq L_{\leqslant i-\Delta k-1}$ and $N(u) \cap L_{i-1} \subseteq V(G_j)$.

We first prove the following claim that we use to find our sequence of separators (S_j).

Claim 2.23. *For n large enough compared to Δ and k and any set of vertices $A \subseteq V(G_1)$ with $|A| \geqslant \log n$, there exists a bag B of T such that B is a $\frac{1}{2}$-balanced separator of A and B is contained in $L_{\leqslant i-\Delta k-1}$.*

Proof. Let T be rooted in a bag that contains v_0. By Theorem 2.15, there is a bag B of T that forms a $\frac{1}{2}$-balanced separator for A (i.e. all connected components of $G_1[A \setminus B]$ are of size at most $|A|/2$). We choose such a bag B of minimum depth (in T). We need to show B is contained in $L_{\leqslant i-\Delta k-1}$.

If B contains v_0, then $B \subseteq G[L_{\leqslant k}]$. Since $i \geqslant 2\Delta k$, we are done in this case.

Suppose now that $v_0 \notin B$ and let B' be the parent of B. By definition, B' is not a $\frac{1}{2}$-balanced separator of A. If B contains a vertex of $L_{\leqslant i-1-k} = V(H_1)$, then its diameter in G_1 is at most k. So either $B \subseteq L_{\leqslant i-\Delta k-1}$ or $B \subseteq L_{>i-(\Delta+1)k-1}$. We are done in the first case, so assume the latter. Since G_1 is connected, $B \cap B' \neq \emptyset$. The same diameter argument gives that $B' \subseteq L_{>i-(\Delta+2)k-1}$. If C is a component of $G_1 \setminus B'$ that does not contain v_0, then the shortest path (of length at most i) from any $v \in C$ must go through B' (at distance at least $i-(\Delta+2)^k-1$ from v_0). In particular, all such components are contained in $N^{((\Delta+2)^k+1)}(B')$ and so the total size is at most $\Delta^{(\Delta+2)k+1}|B'| = O_{k,\Delta}(1)$. For n sufficiently large, this is at most $\frac{1}{2}\log n$.

On the other hand, as B' is not a $\frac{1}{2}$-balanced separator, there exist a component A' of $G_1[A \setminus B']$ with $|A'| > \frac{1}{2}\log n$. We found above that A' must contain v_0. Since B does not contain v_0, A' is contained in the component of $G_1[A \setminus B]$ containing v_0. This yields a contradiction with the fact that B is a $\frac{1}{2}$-balanced separator of A. □

Suppose that we have defined G_j for some $j \geqslant 1$ and let us describe how to define G_{j+1}. If $|G_j| \leqslant \log n$, we ask $\textsc{Query}(u, V(G_j))$ and output $N(u) \cap V(G_j)$. Otherwise, we let S_j be the bag found in Claim 2.23 when applied to $A = V(G_j)$. Then $S_j \subseteq L_{\leqslant i-\Delta k-1}$ and it is a $\frac{1}{2}$-balanced separator of G_j. Since it is a bag of T and contained in H_1, we find the size of the bag is at most $\Delta^k + 1$ and $d_{G_1}(u, v) \leqslant k$ for all $u, v \in S_j$. We ask $\textsc{Query}(N[S_j], u)$ and let G_{j+1} be a component of $G_j \setminus S_j$ that contains a vertex from $\arg\min_{x \in N[S_j]} d_G(x, u)$. Then, we increase j by one and repeat the same procedure.

We now prove the correctness of the algorithm presented above.

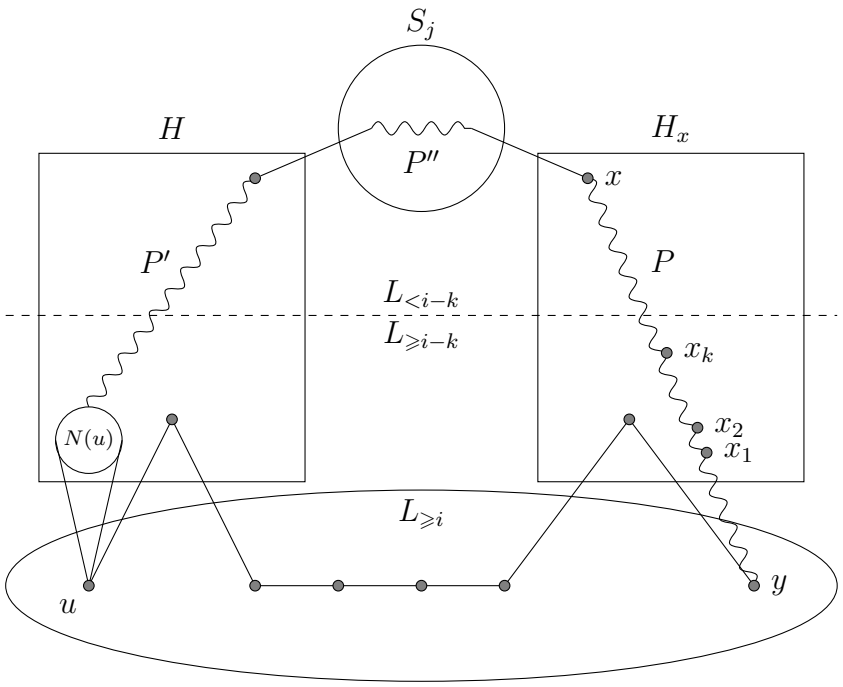


Figure 2.3: This figure depicts a possible configuration in the proof of Claim 2.24 for which we end up with a contradiction by finding a large induced cycle.

.

We first argue that $N(u) \cap L_{i-1}$ is contained in a unique connected component of $G_j \setminus S_j$. Since every separator is included in $L_{\leqslant i-\Delta k-1}$, we find $d_{G_1}(u, S_\ell) \geqslant \Delta k + 1$ for all $\ell \leqslant j$. By Claim 2.22, all vertices in $N(u) \cap L_{i-1}$ are connected via paths in G_1 of length at most Δk and by the observation above, these paths avoid all separators so will be present in a single connected component of $G_j \setminus S_j$. The only thing left to prove is the following claim.

Claim 2.24. *G_{j+1} is the component that contains $N(u) \cap L_{i-1}$.*

Proof. Let $x \in N[S_j]$ such that $d_G(x, u)$ is minimised and let H_x be the connected component of x. We will prove that H_x contains $N(u) \cap L_{i-1}$. In particular, this implies that $G_{j+1} = H_x$ (a priori, G_{j+1} could be a different component for another minimiser than x), so that G_{j+1} contains $N(u) \cap L_{i-1}$, as desired.

Suppose towards a contradiction that $N(u) \cap L_{i-1}$ is instead contained in a different connected component H. We will find an induced cycle of length at least $k + 1$.

Let P be a shortest path in G from x to u.

Let P' be a path from u to S_j with all internal vertices in H. Such a path exists since G_j is connected.

Let P'' be a shortest path in G between a neighbour of x in S_j to the endpoint of P' in S_j. As S_j is a bag contained in H_1, any two vertices in S_j are within distance k in G_1. So $P'' \subseteq L_{\leqslant i-2k-2}$ (we may assume $\Delta \geqslant 4$).

Let y be the first vertex on the path P (from x to u) that lies in L_i (such a vertex must exist since the path does not have internal vertices in S_j by choice of x and since H_x contains no neighbours of u).

Let $x_1, \dots, x_k$ be the k vertices before y in P. Note that none of the x_i can be adjacent to or part of $P' \cup P''$ (since they are in $H_x \cap L_{\geqslant i-k}$). Let G' be the graph obtained from $G[P \cup P'' \cup P']$ by contracting $P'' \cup P' \cup (P \setminus \{x_1, \dots, x_k\})$ to a single vertex p. Note that the selected vertex set is indeed connected and that the resulting graph has vertex set $\{x_1, \dots, x_k, p\}$. Since P was a shortest path in G, the vertex set $\{x_1, \dots, x_k\}$ still induces a path and it suffices to argue about the adjacencies of p. Via edges of P, the vertex p is adjacent to x_1 and x_k. If p was adjacent to x_i for some $i \in [2, k-1]$, then there must be a vertex $y \in P'' \cup P' \cup (P \setminus \{x_1, \dots, x_k\})$ adjacent to x_i. But a case analysis shows this is not possible. (The only vertices adjacent to x_i in P are x_{i+1} and x_{i-1} since P is a shortest path; we already argued that $y \notin P' \cup P''$.) We hence found an induced cycle of length $k+1$, a contradiction. □

We now show how to reconstruct all edges in $E_{i,i}$ incident to u.

Claim 2.25. *If $x \in N(u) \cap L_{i-1}$ and $y \in N(v) \cap L_{i-1}$ for some $uv \in E_{i,i}$, then $d_{G[L_{\leqslant i-1}]}(x, y) \leqslant 2\Delta k$.*

Proof. Since $|N[\{u, v\}]| \leqslant 2\Delta$, it suffices to prove that $d_{G_1}(x, y) \leqslant k$ when the shortest path P in G_1 between x and y avoids other vertices from $N[\{u, v\}]$. As we argued in Claim 2.22 this is true when x or y is a neighbour of both u and v (else we create an induced cycle of length at least $k+1$). So we may assume that $x \in N(u) \setminus N(v)$ and $y \in N(v) \setminus N(u)$. But now $P \cup \{u, v\}$ is an induced cycle of length at least $k+1$. □

Let G_1' be the graph obtained from G_1 by adding the vertices in L_i and the edges in $E_{i,i-1}$. Our algorithm already reconstructed G_1'. If $uv \in E_{i,i}$, then applying Claim 2.25 to vertices $x, y \in L_{i-1}$ on the shortest paths from u, v to the root v_0 respectively, we find that $d_{G_1'}(u, v) \leqslant 2\Delta k + 2$. For each $u \in L_i$, we ask $\textsc{Query}(u, N_{G_1'}^{\leqslant 2\Delta k+2}(u) \cap L_i)$ and we record the vertices v for which the response is 1. Those are exactly the vertices adjacent to u. Per vertex $u \in L_i$, this takes at most $\Delta^{2\Delta k+3}$ queries.

The query complexity of reconstructing $E_{i-1,i}$ is $O_{k,\Delta}(\log n |L_i|)$ as there are at most $\log n$ iterations (using the fact that the $(S_j)_j$ are $\frac{1}{2}$-balanced separators) and in each iteration we do $O_{k,\Delta}(\log n)$ queries per vertex $u \in L_i$. In order to reconstruct $E_{i,i}$, we use

$O_{k,\Delta}(1)$ queries per vertex of L_i. Therefore, the total query complexity of the algorithm is $\sum_i O_{k,\Delta}(|L_i| \log n) = O_{k,\Delta}(n \log n)$. □

Using a similar technique while putting in additional effort to optimised the dependency on Δ in the case of (3-)chordal graphs, we obtain an algorithm tight up to a $O(\log \Delta)$ factor. We omit the proof here, as it is similar to the proof of Theorem 2.9, but invovles a considerable amount of technical details.

Theorem 2.8 ([25]). *There exists a deterministic algorithm to reconstruct the class of connected n-vertex chordal graphs of maximum degree at most $\Delta \leqslant \sqrt{\log n}$ using $O(\Delta n \log n)$ distance queries.*

2.4 Reconstruction in $G(n,p)$

We will now focus on reconstructing $G \sim G(n,p)$ for $\frac{2000 \log n}{n} \leqslant p \leqslant n^{-\frac{1}{2}-\varepsilon}$. Note that in this setting the average degree of G, often denoted Δ in this chapter, is equal to $(1+o(1))np$ with high probability.

Theorem 2.12 ([19]). *For any $\varepsilon \geqslant 0$, and every $n \in \mathbb{N}$, for every $\frac{2000 \log n}{n} \leqslant p \leqslant n^{-\frac{1}{2}-\varepsilon}$, there exists an algorithm that reconstructs $G \sim G(n,p)$ using $O(\Delta^2 n \log n)$ queries in expectation, where $\Delta = (n-1)p$ is the expected average degree of G.*

Before proving Theorem 2.12, let us desrcibe here the algorithm we employ, introduced by Mathieu and Zhou [141] and also used in [126]. Similar to previous works, the correctness of the algorithm is straightforward, with the main challenges lying in its complexity analysis.

A simple algorihtm Consider a random set $S \subseteq V(G)$ of fixed size $|S| = s$. First, the algorithm asks $\textsc{Query}(S, V(G))$. From these queries, it computes the *pseudo-edges* of G define as

$$\tilde{E} := \{\{u,v\} \subseteq V(G) \mid \forall s \in S, |d(s,u) - d(s,v)| \leqslant 1\}.$$

The algorithm asks $\textsc{Query}(\tilde{E})$ and finishes. Note that this algorithm always completely reconstructs the graph. If there exists s such that $|d(s,u) - d(s,v)| \geqslant 2$ then $uv \notin E(G)$, therefore $\tilde{E}$ is a superset of $E(G)$. The number of queries asked is at most $ns + |\tilde{E}|$. Therefore the challenge in proving precise upper bounds lies in showing that even for small values of s, $|\tilde{E}|$ does not become excessively large.

Discussion This algorithm is extremely simple and natural, similarly to previous works, the correctness of the algorithm is straightforward, with the main challenges lying in its complexity analysis. Theorem 2.12 was found independently from [126], the primary novel aspect, compared to [126], is deriving bounds even for very sparse graphs where the diameter grows with n and $p \leqslant \log^2 n/n$. This requires a detailed study of the interactions within the k-neighbourhoods of non-adjacent vertices. More specifically, given two vertices $u, v \in V(G)$, and an integer k, we derive precise bounds for the number of vertices in the set $\{x \in V(G) \mid d(u,x) = k \wedge d(v,x) \in [k, k+1]\}$. Our method might also be applicable for establishing bounds for other types of distance profiles between pairs of vertices. That is, given two vertices u, v and two distances i and j, computing tight bounds on the distribution of vertices satisfying $\{x \mid d(u,x) = i \wedge d(v,x) = j\}$.

2.4.1 Proof of Theorem 2.12

From now on, and for the rest of the subsection let us set $\frac{2000 \log n}{n} \leqslant p \leqslant n^{-1/2-\varepsilon}$. We consider $G \sim G(n,p)$. Let $\Delta = p(n-1)$. Note that Δ is the expected average degree in G. A key notion for the proof is vertices that are called *witnesses* defined below. Intuitively, a witness x of a non-edge $uv \notin E(G)$ is a vertex such that the answers to $\textsc{Query}(\{x\}, \{u, v\})$ are enough to conclude that $uv \notin E(G)$.

Definition 2.26. *We say that $s \in V(G)$ is a witness of the non-edge $uv \notin E(G)$ if $|d(s,u) - d(s,v)| \geqslant 2$. We denote W_{uv} the set of witnesses of the non-edge uv.*

Let us first prove the following key lemma, stating that almost all pairs of vertices have a large number of witnesses.

Lemma 2.27 (Main Lemma)**.** *The set $\mathcal{E} = \{uv \notin E(G) \mid |W_{uv}| \geqslant \frac{n}{257\Delta^2}\}$ has size at least $\binom{n}{2} - O(n\Delta^2)$ with probability $1 - o(n^{-1})$.*

Proof. Since we aim to prove the statement with a probability of $1 - o(n^{-1})$, we can apply a union bound to a constant number of events, each having a probability of $1 - o(n^{-1})$. For example, we will assume deterministic conditioning on the fact that Lemma 2.18 holds. We will also use similar reasoning throughout the proof, in which case, we will say that we assume that an event (here Lemma 2.18) holds deterministically. We first make the following easy claim:

Claim 2.28. *There are at most $\frac{9}{4}n\Delta^2$ pairs of vertices $u, v \in V(G)$ satisfying $d(u,v) \leqslant 2$ with probability $1 - o(n^{-1})$.*

Proof. For a fixed $u \in V(G)$, from Lemma 2.18, $|N^2(u)| \leqslant \left(\frac{3}{2}\Delta\right)^2 = \frac{9}{4}\Delta^2$. The claim follows from double counting. □

Intuitively, such pairs of vertices are too close and therefore may have plenty of non-witnesses, but Claim 2.28 ensures that there are $O(n\Delta^2)$ such pairs, therefore, we can assume that none of these pairs is in $\mathcal{E}$, and focus on proving that the set $\{uv \notin E(G) \mid d(u,v) \geqslant 3 \wedge |W_{uv}| \geqslant \frac{n}{257\Delta^2}\} \subseteq \mathcal{E}$ is large enough. We will in fact prove $\{uv \notin E(G) \mid d(u,v) \geqslant 3\} \subseteq \mathcal{E}$ with probability $1 - o(n^{-1})$.

To do so, we consider a uniformly chosen pair of vertices $uv \in \binom{V(G)}{2}$ and we prove that either $d(u,v) \leqslant 2$ or $|W_{uv}| \geqslant (1/2 - o(1))n/\Delta^2$ with probability $1 - o(n^{-3})$. Let us denote by $N^k(u,v)$ the combined k-neighbourhood of u and v,

$$N^k(u,v) = \{x \in V(G) \mid \min(d(u,x), d(v,x)) = k\}.$$

Our approach sequentially reveals edges of G by spheres of increasing radius around u and v. By that, we mean that we first reveal the edges from $\{u,v\}$ to $N(u,v)$ then from $N(u,v)$ to their neighbors outside of $\{u,v\}$, ... until we reveal all the edges. Therefore, at step k, we reveal the edges xy where $x \in N^k(u,v)$ and $y \in N^{k+1}(u,v)$, but also the edges xy where both x and y are in $N^k(u,v)$. This sequential procedure is equivalent to sampling $G \sim G(n,p)$, but simplifies studying the evolution of the structure of $N^k(u,v)$ when k grows. We define a partition of $N^k(u,v)$ into two sets A_k and B_k. Intuitively, A_k will contain a high density of witnesses of the pair u,v. On the contrary, we will add to B_k any vertex that could potentially be a non-witness of uv. We will define B_k as the union of three subsets, $\mathcal{B}_1^k, \mathcal{B}_2^k$ and $\mathcal{B}_3^k$, represented in Fig. 2.4. We give a formal definition of each subset below, and then of A_k and B_k, as well as a short informal explanation of why vertices in these parts could be non-witnesses:

- $\mathcal{B}_1^k = N(B_{k-1})$: Suppose $x \in B_{k-1}$ is a non-witness, for example $d(x,u) = d(x,v) = k-1$, then any neighbour y of x in $N^k(u,v)$ satisfies $d(y,u) = d(y,v) = d(x,u)+1 = k$ and is a non-witness of uv, therefore $N(x)$ should be in B_k.

- $\mathcal{B}_2^k = N(\{x \in A_{k-1} \mid N(x) \cap N^{k-1}(u,v) \neq \emptyset\})$: Consider $x,y \in A_{k-1}$ such that $xy \in E(G)$, and suppose that before the xy edge is revealed, both x and y have a distance profile that makes them potential witnesses of uv. One possible case is $d_{G\setminus\{xy\}}(x,u) = d_{G\setminus\{xy\}}(y,v) = k-1$, $d_{G\setminus\{xy\}}(x,v) > k$ and $d_{G\setminus\{xy\}}(y,u) > k$. When the edge xy is revealed the distance profiles are modified: $d(x,u) = d(y,v) = k-1$ and $d(x,v) = d(y,u) = k$. Both x and y become non-witnesses. Any neighbour $z \in N(x) \cup N^k(u,v)$ satisfy $d(z,u) = k$ and $k \leqslant d(z,v) \leqslant d(z,x) + d(x,v) = k+1$. A symmetric reasoning for y implies that every vertex in $(N(x) \cup N(y)) \cap N^k(u,v)$ is not a witness of uv.

- $\mathcal{B}_3^k = \left\{x \in N^k(u,v) \middle| |N(x) \cap A_{k-1}| \geqslant 2\right\}$: Suppose x,y are two distinct witnesses of uv. Again, one possible case is $d(x,u) = d(y,u) = k-1$, $d(x,v) > k$ and $d(y,u) > k$. Suppose they share a neighbour z in $N^k(u,v)$. Now the distance profile of z is

$d(z,u) = d(x,u) + 1 = k$ and $d(z,v) = d(y,v) + 1 = k$, therefore z is a non-witness of uv.

Let $A_1 = N(u,v)$ and $B_1 = \emptyset$ we define recursively, for $k \in \mathbb{N}$,

$$B_k = N^k(u,v) \cap (\mathcal{B}_1^k \cup \mathcal{B}_2^k \cup \mathcal{B}_3^k) \qquad A_k = N^k(u,v) \setminus B_k.$$

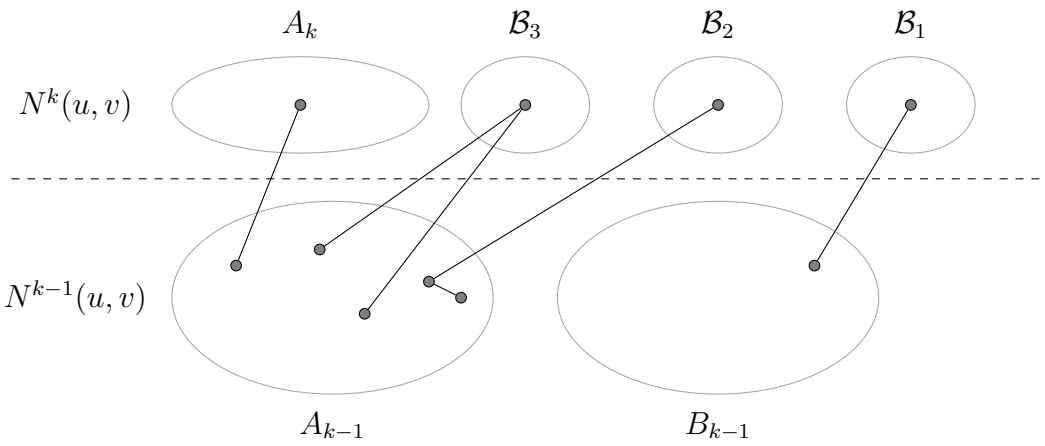


Figure 2.4: Representation of the three parts $\mathcal{B}_1^k, \mathcal{B}_2^k, \mathcal{B}_3^k$ that compose B_k

The proof is structured around three claims. Claim 2.29 ensures that at least a fourth of the vertices of A_k are witnesses of u, v with high probability.

Claim 2.29. *For any k such that $|N^k(u,v)| \leqslant 3|A_k| \leqslant \frac{n}{16\Delta}$, we have either $d(u,v) \leqslant 2$ or $|A_k \cap W_{uv}| \geqslant \frac{1}{4}|A_k|$ with probability $1 - o(n^{-3})$.*

Claim 2.30 and Claim 2.31 control the size of B_k (and therefore A_k) inductively. Specifically, Claim 2.30 addresses the case where $|B_k|$ is small. In such cases, we allow a constant factor increase in the upper bound of $|B_{k+1}|$. Conversely, Claim 2.31 handles larger $|B_k|$ with a tighter concentration and increase factor close to 1.

Claim 2.30. *For any constant $\beta \geqslant 0$, for n large enough and for any $k \in \mathbb{N}$ if $|N^{k-1}(u,v)| \leqslant n/\Delta^2$ then*

$$|B_{k-1}| \leqslant \beta |N^{k-1}(u,v)| \implies |B_k| \leqslant \left(3\beta + \frac{1}{26}\right) |N^k(u,v)|$$

with probability at least $1 - n^{-4}$.

Claim 2.31. *For n large enough, for any $k \geqslant 1$, if $|N^{k-1}(u,v)| \leqslant n/\Delta^2$ and $|B_{k-1}| \geqslant \Delta^2/4$ then*

$$|B_{k-1}| \leqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{k-1}|N^{k-1}(u,v)| \implies |B_k| \leqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{k}|N^{k}(u,v)|$$

with probability at least $1-n^{-4}$.

The factor $(1+\frac{30}{\log n})^k$ in Claim 2.31 is just a product of the induction. It is important to note that the range of k of interest in this claim is upper bounded by the diameter of G, which means $k = O(\log n/\log\log n)$. In this range, we have $(1+\frac{30}{\log n})^k \leqslant e^{30k/\log n} = e^{o(1)} = 1+o(1)$.

Let us first prove that the lemma follows from the three claims. We assume that Claim 2.29, Claim 2.30 and Claim 2.31 hold deterministically. We will now show that for any k such that $|N^{k-1}(u,v)| \leqslant n/\Delta^2$, we have $|A_k| \geqslant (1/2-o(1))|N^{k-1}(u,v)|$. Since A_k, B_k forms a partition of $N^k(u,v)$, it is equivalent to show that $|B_k| \leqslant (1/2+o(1))|N^{k-1}(u,v)|$. Let us consider $i \in \mathbb{N}$ to be the first step such that B_{i+1} is non-empty. Note that $i \geqslant 1$ as $B_1 = \emptyset$. By Claim 2.30 applied with $\beta = 0$, we have that $1 \leqslant |B_{i+1}| \leqslant \frac{1}{26}|N^{i+1}(u,v)|$. Applying Claim 2.30 again for B_{i+1}, and then B_{i+2} with first $\beta = 1/26$ and then $\beta = 1/2$, we obtain the following,

$$|B_{i+2}| \leqslant \frac{2}{13}|N^{i+2}(u,v)| \quad \text{and} \quad |B_{i+3}| \leqslant \frac{1}{2}|N^{i+3}(u,v)|.$$

By maximality of i, B_{i+1} is non-empty. Therefore we can consider $z \in B_{i+1}$, by definition of $\mathcal{B}_1^{i+2}$ we have $N^{i+2}(u,v) \cap N(z) \subseteq \mathcal{B}_1^{i+2} \subseteq B_{i+2}$. Remember that we can suppose $N^{i+2}(u,v) \leqslant n/\Delta^2$. The probability that a vertex $x \notin N^{i+1}(u,v)$ is a neighbour of x is p. Therefore $\mathbb{E}[|N^{i+2}(u,v) \cap N(z)|] = (1-1/\Delta^2)pn = (1-o(1))\Delta$. We apply a Chernoff's bound (Lemma 0.1) with parameter $\delta = \frac{1}{2}$ and $\mu = (1-o(1))\Delta$ to obtain that $|N^{i+2}(u,v) \cap N(z)| \geqslant \Delta/2$ with probability $1-o(n^3)$. Applying similar reasoning, we can argue that $|N^{i+3}(u,v) \cap N^2(z)| \geqslant \Delta^2/4$ with probability $1-o(n^3)$.

Recall that $N^{i+3}(u,v) \cap N^2(z) \subseteq B_{i+3}$, therefore, $|B_{i+3}| \geqslant \Delta^2/4$ whp, and the conditions are met to apply Claim 2.31 for $k \geqslant i+3$, as long as $|N^{k-1}(u,v)| \leqslant n/\Delta^2$. Let ℓ denote the largest integer such that $|N^{\ell}(u,v)| \leqslant \frac{n}{32\Delta^2}$ is satisfied. Note that as $\frac{2000\log n}{n} \leqslant p \leqslant n^{-1/2-\varepsilon}$, ℓ is indeed well-defined whp. Moreover, it is upper bounded by the diameter of G, therefore

$\ell = O(\log n / \log\log n)$ whp. Applying Claim 2.31 we obtain,

$$\begin{aligned}|A_\ell| &\geqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{\ell}|N^\ell(u,v)| \\ &\geqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{O(\frac{\log n}{\log\log n})}\frac{n}{32\Delta^2} \\ &\geqslant \left(\frac{1}{64}+o(1)\right)\frac{n}{\Delta^2}.\end{aligned}$$

The second inequality holds as $k \mapsto (1+\frac{30}{\log n})^k|N^k(u,v)|$ is increasing whp for $k \leqslant \ell$. It is now sufficient to apply Claim 2.29 for $k = \ell$ to conclude that $|W_{uv}| \geqslant |A_k|/4 \geqslant (1/256 - o(1))n/\Delta^2$ with probability $1 - o(n^{-3})$. By using a union bound over all such pairs. We obtain that all pairs $uv \notin E(G)$, such that $d(u,v) \geqslant 3$ satisfy $|W_{uv}| \geqslant (1/256 + o(1))n/\Delta^2$. Finally, using Claim 2.28 we conclude that $|\mathcal{E}| \geqslant |\{uv \notin E(G) | d(u,v) \geqslant 3\}| \geqslant \binom{n}{2} - O(n\Delta^2)$ with probability $1 - o(n^{-1})$.

We will now prove Claim 2.29 - Claim 2.31. We start by Claim 2.31 as it encapsulates the main ideas of the proof.

Claim 2.31. *For n large enough, for any $k \geqslant 1$, if $|N^{k-1}(u,v)| \leqslant n/\Delta^2$ and $|B_{k-1}| \geqslant \Delta^2/4$ then*

$$|B_{k-1}| \leqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{k-1}|N^{k-1}(u,v)| \implies |B_k| \leqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{k}|N^k(u,v)|$$

with probability at least $1 - n^{-4}$.

Proof of Claim 2.31. We will upper bound the size of the three different parts $\mathcal{B}_1^k, \mathcal{B}_2^k$ and $\mathcal{B}_3^k$ that compose B_k independently:

- $\mathcal{B}_1^k = N(B_{k-1})$: Consider a vertex $x \notin N^{k-1}(u,v)$, the probability that $x \in N(B_{k-1})$ can be written as:

$$\begin{aligned}\mathbb{P}(x \in N(B_{k-1})) &= \mathbb{P}(\mathrm{Bin}(|B_{k-1}|, p) \geqslant 1) \\ &= 1 - (1-p)^{|B_{k-1}|} \\ &= (1-o(1))p|B_{k-1}|\end{aligned}$$

  therefore,

$$\mathbb{E}[|\mathcal{B}_1^k|] = (1-o(1))p|B_{k-1}|(n - |B_{k-1}|) = (1-o(1))\Delta|B_{k-1}| \leqslant \Delta|B_{k-1}|.$$

By Chernoff's bound (Lemma 0.1) with $\delta = \frac{1}{\log n}$, we obtain that

$$\begin{aligned}\mathbb{P}\left(|\mathcal{B}_1^k| \geqslant \left(1+\frac{1}{\log n}\right)\Delta|B_{k-1}|\right) &\leqslant e^{-(1-o(1))\Delta|B_{k-1}|/(3\log^2 n)} \\ &\leqslant e^{-(1-o(1))\Delta^3/(12\log^2 n)} \\ &\leqslant n^{-5}\end{aligned}$$

for n large enough.

- $\mathcal{B}_2^k = N(\{x \in A_{k-1} \mid N(x) \cap N^{k-1}(u,v) \neq \emptyset\})$: This set is contained in the neighbourhood of non-isolated vertices in the graph $H := G[N^{k-1}(u,v)]$. Note that the vertex set $N^k(u,v)$ is defined independently of the distribution of the edges in H. Therefore $H \sim G(|N^{k-1}(u,v)|, p)$. Let N denote $|N^{k-1}(u,v)|$. By assumption we have $|B_{k-1}| \leqslant N \leqslant n/\Delta^2$. Therefore $p = \frac{\Delta}{n} \leqslant \frac{1}{N\Delta}$. We can apply Lemma 2.19 to H and deduce that $|\{x \in A_{k-1} \mid N(x) \cup N^{k-1}(u,v) \neq \emptyset\}| \leqslant 4N/\Delta$ with probability at least
$$1 - 2e^{-N/3\Delta} \geqslant 1 - 2e^{-\Delta/12} \geqslant 1 - n^{-5}.$$
Applying Lemma 2.18 we obtain,
$$\mathbb{P}\left(|\mathcal{B}_2^k| \geqslant 6|N^{k-1}(u,v)|\right) \leqslant n^{-5}.$$

- $\mathcal{B}_3^k = \left\{x \in N^k(u,v) \middle| |N(x) \cap A_{k-1}| \geqslant 2\right\}$: Given a vertex $x \in N^k(u,v)$, consider M_x the event: $|N(x) \cap A_{k-1}| \geqslant 2$. If $x \in N^k(u,v)$ has a neighbour in B_{k-1} then x has already been counted in $\mathcal{B}_1^k$. Therefore we can suppose x has a neighbour in A_{k-1}. The probability for such an x to satisfy M_x can be written as:
$$\begin{aligned}\mathbb{P}(M_x \mid |N(x) \cap A_{k-1}| \geqslant 1) &= \mathbb{P}(|N(x) \cap A_{k-1}| \geqslant 2 \mid |N(x) \cap A_{k-1}| \geqslant 1) \\ &= \mathbb{P}(\mathrm{Bin}(|A_{k-1}| - 1, p) \geqslant 1) \\ &\leqslant 1 - (1-p)^{|A_{k-1}|-1} \\ &\leqslant 1 - \left(1 - \frac{\Delta}{n}\right)^{n/\Delta^2} \\ &\leqslant 1/\Delta\end{aligned}$$
Note that the M_x are mutually independent, therefore using Chernoff's bounds (Lemma 0.1), over $|N^k(u,v)| \geqslant \Delta^2/4$ i.i.d. variables with $\delta = 1$ and $\mu \leqslant |N^k(u,v)|/\Delta$, we deduce,
$$\mathbb{P}\left(|\mathcal{B}_3^k| \geqslant 2|N^k(u,v)|/\Delta\right) \leqslant n^{-5}.$$

Summing and using a union bound on the three events described above we get that,

$$\mathbb{P}\left(|B_k| \leqslant \left(1+\frac{1}{\log n}\right)\Delta|B_{k-1}| + 6|N^{k-1}(u,v)| + \frac{2|N^k(u,v)|}{\Delta}\right) \geqslant 1-3n^{-5}.$$

Let us rewrite this inequality to fit the statement of the theorem.

$$\begin{aligned}
|B_k| &\leqslant \left(1+\frac{1}{\log n}\right)\Delta|B_{k-1}| + 6|N^{k-1}(u,v)| + \frac{2|N^k(u,v)|}{\Delta} \\
&\leqslant \left(1+\frac{1}{\log n}\right)\left(1+\frac{30}{\log n}\right)^{k-1}\frac{\Delta}{2}|N^{k-1}(u,v)| + 6|N^{k-1}(u,v)| + \frac{2|N^k(u,v)|}{\Delta} \\
&\leqslant \left(1+\frac{1}{\log n}+\frac{12}{\Delta}\right)\left(1+\frac{30}{\log n}\right)^{k-1}\frac{\Delta}{2}|N^{k-1}(u,v)| + \frac{2|N^k(u,v)|}{\Delta} \\
&\leqslant \left[\left(1+\frac{2}{\log n}+\frac{12}{\Delta}\right)\left(1-\frac{1}{\log n}\right)^{-1}+\frac{2}{\Delta}\right]\left(1+\frac{30}{\log n}\right)^{k-1}\frac{1}{2}|N^k(u,v)| \\
&\leqslant \left(1+\frac{30}{\log n}\right)\left(1+\frac{30}{\log n}\right)^{k-1}\frac{1}{2}|N^k(u,v)| \\
&\leqslant \frac{1}{2}\left(1+\frac{30}{\log n}\right)^{k}|N^k(u,v)|,
\end{aligned}$$

where we use the claim hypothesis to go from line 1 to 2. From line 3 to 4, we use the following fact, which can be derived from a Chernoff's bound (Lemma 0.1) in the same way as the above upper bound on $N(B_{k-1})$: $|N^k(u,v)| \geqslant (1-\frac{1}{\log n})\Delta|N^{k-1}(u,v)|$. □

Let us now discuss our bound on B_k for the small values of k, when $|B_k| \leqslant \Delta^2/4$. The proof follows the same guideline as Claim 2.31.

Claim 2.30. *For any constant $\beta \geqslant 0$, for n large enough and for any $k \in \mathbb{N}$ if $|N^{k-1}(u,v)| \leqslant n/\Delta^2$ then*

$$|B_{k-1}| \leqslant \beta|N^{k-1}(u,v)| \implies |B_k| \leqslant \left(3\beta+\frac{1}{26}\right)|N^k(u,v)|$$

with probability at least $1-n^{-4}$.

Proof of Claim 2.30. What differs from Claim 2.31 is only that Chernoff's inequality yields a weaker concentration. But the statement is tailored to absorb this lost concentration into the error factor $(3\beta+\frac{1}{100})$.

- $\mathcal{B}_1^k = N(B_{k-1})$: We can consider B_{k-1} to be non-empty (otherwise $N(B_{k-1}) = \emptyset$). Again, a Chernoff bound (Lemma 0.1) with $\delta = 1$ and $\mu = (1 + o(1))\Delta|B_{k-1}|$ implies that,

$$\mathbb{P}\left(|\mathcal{B}_1^k| \geqslant 2\Delta|B_{k-1}|\right) \leqslant e^{(1+o(1))\Delta/3} \leqslant n^{-5}.$$

  Note that $|N^k(u,v)| \leqslant \frac{3}{2}|N^{k-1}(u,v)|$ by Lemma 2.18. Therefore,

$$|\mathcal{B}_1^k| \leqslant 2\Delta|B_{k-1}| \leqslant 2\Delta\beta|N^{k-1}(u,v)| \leqslant 3\beta|N^k(u,v)|.$$

- $\mathcal{B}_2^k = N(\{x \in A_{k-1} \mid N(x) \cap N^{k-1}(u,v) \neq \emptyset\})$: Instead of directly bounding the size of $\mathcal{B}_2^k$ we use the fact that $|\mathcal{B}_2^k|$ is upper bounded by two times the number of edges in the graph $G[N^{k-1}(u,v)]$. Let us denote this random variable by $M := |E(G[N^{k-1}(u,v)])|$ and $\mu = \mathbb{E}[M]$. Note that $\mu = O(p|N^{k-1}(u,v)|^2) \ll |N^{k-1}(u,v)|$ by assumption. Therefore we can fix $\delta_2 \geqslant 1$ to satisfy $(1+\delta_2)\mu = \frac{1}{104}|N^{k-1}(u,v)|$ and apply Chernoff's bounds

$$\begin{aligned}\mathbb{P}\left(|\mathcal{B}_2^k| \geqslant \frac{\Delta}{52}|N^{k-1}(u,v)|\right) \leqslant \mathbb{P}\left(M \geqslant \frac{1}{104}|N^{k-1}(u,v)|\right) &\leqslant e^{-(1+o(1))\delta_2\mu/3} \\ &\leqslant e^{-\frac{1+o(1)}{312}|N^1(u,v)|} \\ &\leqslant n^{-5}.\end{aligned}$$

- $\mathcal{B}_3^k = \{x \in N^k(u,v) \mid |N(x) \cap A_{k-1}| \geqslant 2\}$. From the same reasoning as the proof of Claim 2.31, we know that $\mathbb{P}(|N(x) \cap A_{k-1}| \geqslant 2 \mid x \in A_k) \leqslant 1/\Delta$. We can also use Chernoff's bounds (Lemma 0.1). Fix $\delta_3 \geqslant 1$ which satisfies $(1+\delta_3)\mathbb{E}(|\mathcal{B}_2^k|) = \frac{\Delta}{104}|N^1(u,v)| \leqslant \frac{1}{52}|N^2(u,v)|$. As for $\mathcal{B}_2^k$ we have,

$$\begin{aligned}\mathbb{P}(|\mathcal{B}_3^k| \geqslant \frac{\Delta}{104}|N^1(u,v)|) &\leqslant e^{-\Delta^2/312} \\ &\leqslant n^{-5}.\end{aligned}$$

Using a union bound on the three events described above, we obtain:

$$\mathbb{P}\left(|B_k| \leqslant 3\beta|N_k(u,v)| + \frac{2|N^k(u,v)|}{52}\right) \geqslant 1 - 3n^{-5}.$$

□

Finally, we are left to prove that A_k contains a large number of witnesses of the pair u, v.

Claim 2.29. *For any k such that $|N^k(u,v)| \leqslant 3|A_k| \leqslant \frac{n}{16\Delta}$, we have either $d(u,v) \leqslant 2$ or $|A_k \cap W_{uv}| \geqslant \frac{1}{4}|A_k|$ with probability $1 - o(n^{-3})$.*

Proof of Claim 2.29. We will prove that, if $d(u,v) \geqslant 3$, then any vertex $x \in A_k$ such that $N(x) \cap N^k(u,v) = \emptyset$ is a witness of u, v. To show that this is a sufficient claim, we can reasong as in Claim 2.30 and upper bound $|\{x \in A_k | N(x) \cap N^k(u,v) \neq \emptyset\}|$ using Lemma 2.19 on $H := G[N^k(u,v)] \sim G(N,p)$ where $n := |N^k(u,v)|$ and $D := 16$, to get that at most $4N/D = N/4 \leqslant \frac{3}{4}|A_k|$ vertices of A_k do not satisfies $N(x) \cap N^k(u,v) = \emptyset$.

Suppose that $d(u,v) \geqslant 3$, we prove by induction on k that for any $x \in A_k$, if $N(x) \cap N^k(u,v) = \emptyset$ then x is a witness of u, v. It is true for $k = 1$ as by assumption $d(u,v) \geqslant 3$, therefore any vertex $x \in N(u,v)$ satisfying $N(x) \cap N(u,v) = \emptyset$ is not on a length 3 path between u and v, and must satisfy either $d(u) = 1$ and $d(v) \geqslant 3$ or symmetrically $d(v) = 1$ and $d(u) \geqslant 3$. Assume that the statement holds up to A_{k-1}. Consider $x \in A_k$ and the unique $y \in A_{k-1}$ such that $xy \in E(G)$. Such a y exists by construction of A_k. Note that the definition $\mathcal{B}_2^k$ ensures that if $x \in A_k$ then y satisfies the induction hypothesis: $N(y) \cap N^{k-1}(u,v) = \emptyset$. Therefore, we can consider up to symmetry that $d(y,u) = k-1$ and $d(y,v) \geqslant k+1$. Thus $d(x,u) = k$ and $d(y,v) \geqslant k+2$ as its unique neighbour in $N^{k-1}(u,v) \cup N^k(u,v)$ is y. Therefore, x is a witness of uv. □

□

Theorem 2.12. *For any $\varepsilon \geqslant 0$, and every $n \in \mathbb{N}$, for every $\frac{2000 \log n}{n} \leqslant p \leqslant n^{-\frac{1}{2}-\varepsilon}$, there exists an algorithm that reconstructs $G \sim G(n,p)$ using $O(\Delta^2 n \log n)$ queries in expectation, where $\Delta = (n-1)p$ is the expected average degree of G.*

Proof. Recall that $\Delta = (n-1)p$. We will prove an upper bound on the expected query complexity of $O(\Delta^2 n \log n)$. Let $\overline{E} := \binom{V(G)}{2} \setminus E(G)$ be the set of non-edges of G. The algorithm proceeds as follows. First, we consider a randomly sampled set S of vertices where $|S| = 10^3 \Delta^2 \log n$. We ask $\textsc{Query}(V, S)$. We then deduce

$$D = \{\{u,v\} \in \binom{V(G)}{2} \mid \exists s \in S, |d(s,u) - d(s,v)| \geqslant 2\},$$

and query all pairs in $\binom{V(G)}{2} \setminus D$. Note that $D \subseteq \overline{E}$ by definition, therefore the correctness of this algorithm is direct.

We now prove that $\binom{[n]}{2} \setminus D = O(\Delta^2 n)$ with probability $1 - o(n^{-1})$. Consider the set $D = \{uv \in \binom{[n]}{2} \mid |W_{uv}| \geqslant \frac{n}{257\Delta^2}\}$. Lemma 2.27 ensures us that $|D| = \binom{n}{2} - O(\Delta^2 n)$ with probability $1 - o(n^{-1})$. Assume that Lemma 2.27 holds deterministically. We compute the probability that a fixed pair $uv \in D$ has no witness in S.

$$\begin{aligned}\mathbb{P}(S \cap W_{uv} = \emptyset) &\leqslant \left(1 - \frac{n}{257n\Delta^2}\right)^{10^3\Delta^2 \log(n)} \\ &\leqslant e^{-\frac{10^3}{257}\log(n)} \\ &\leqslant o(n^{-3}).\end{aligned}$$

If we union bound over the events $S \cap W_{uv} \neq \emptyset$ for all pairs of vertices, we conclude that $|\binom{[n]}{2} \setminus \tilde{E}| = O(\Delta^2 n)$ with probability $1 - o(n^{-1})$, in which case, our algorithm uses $n|S| + O(\Delta^2 n) = O(\Delta^2 n \log n)$ queries. The worst-case query complexity of the algorithm being $O(n^2)$ we obtain that the expected number of queries done by the algorithm is $(1 - o(n^{-1}))\Delta^2 n \log n + o(n^{-1})n^2 = O(\Delta^2 n \log n)$. □

2.5 Query complexity lower bounds

In this section we prove that the algorithms presented in Section 2.3.1 are optimal in terms of the dependency on n and Δ, even when randomisation is allowed. The theorem below also imply that the results in Section 2.3.2 are optimal in term of there dependency in n.

Theorem 2.14. *Let $\Delta \geqslant 2$ and $n = 2c\Delta^k$ be integers, where $c \in [1, \Delta)$ and $k \geqslant 50(c \ln c + 3)$ is an integer. Any randomised algorithm requires at least $\frac{1}{50}\Delta n \log_\Delta n$ queries in expectation to reconstruct n-vertex trees of maximum degree $\Delta + 1$.*

Note that, for constant c, Δ could even be a small polynomial in n. The proof that we develop is flexible, and allows us to derive easily new lower bounds for similar reconstruction questions presented in Section 2.5.4.

For the rest of this section any "algorithm" is allowed to be randomised unless specified to be deterministic.

2.5.1 Reconstructing functions from the coordinate oracle

In order to prove the lower bound we reduce to a natural function reconstruction problem that could be of independent interest. Let $\Delta \geqslant 3$, $k \geqslant 1$ and $n = c\Delta^k$ be integers, where $c \in [1, \Delta)$.

Let $A = [n]$ and $B = [\Delta]^k$. Suppose that $f : A \to B$ is an unknown function that we want to reconstruct. For $b \in B$ and $1 \leqslant i \leqslant k$, we write b_i for the value of the ith coordinate of b.

The *coordinate oracle* can answer the following two types of queries:

- **Type 1**. $\textsc{Query}_1^c(a, b, i)$ for $a \in A$, $b \in [\Delta]$ and $i \in [k]$ answers **YES** if $f(a)_i = b$ and **NO** otherwise.
- **Type 2**. $\textsc{Query}_2^c(a, a', i)$ for $a, a' \in A$ and $i \in [k]$ answers **YES** if $f(a)_i = f(a')_i$ and **NO** otherwise.

In the case of the coordinate oracle, we will count the number of queries for which the answer is **NO** instead of the number of queries.

We say that $f : A \to B$ is a *balanced function* if for every $b \in B$, $|f^{-1}(b)| = c$ for some integer $c \geqslant 1$.

Our main result on function reconstruction from a coordinate oracle is the following.

Theorem 2.32. *Let $\Delta \geqslant 3$, $c \leqslant \Delta - 1$ and $k \geqslant 50(c \ln c + 2)$ be positive integers and let $n = c\Delta^k$. Any algorithm reconstructing $f : [n] \to [\Delta]^k$ using the coordinate oracle, in the special case where f is known to be a balanced function, has at least $\frac{1}{11}\Delta nk$ queries answered* **NO** *in expectation.*

In order to prove Theorem 2.32, we first study the query complexity in the general case, when no restriction is put on f. Using Yao's minimax principle [183], studying the expected complexity of a randomised algorithm can be reduced to studying the query complexity of a deterministic algorithm on a randomised input.

Lemma 2.33 (Corollary of Yao's minimax principle [183])**.** *For any distribution D on the inputs, for any randomised algorithm M, the expected query complexity of M is at least the average query complexity of the best deterministic algorithm for input distribution D.*

We will apply Yao's principle for D the uniform distribution and the query complexity measuring the number of queries answered **NO**. We combine this with the following lemma.

Lemma 2.34. *Let n, k and Δ be integers. For any deterministic algorithm R using the coordinate oracle and $f : [n] \to [\Delta]^k$ sampled u.a.r., the probability that R reconstructs f in at most $\frac{1}{10}\Delta nk$ queries answered* **NO** *is at most $e^{-\frac{1}{50}nk}$.*

We first deduce our main theorem on function reconstruction from the two lemmas above.

Proof of Theorem 2.32. Let M be a deterministic algorithm that reconstructs balanced functions using the coordinate oracle. We first extend M to an algorithm $\widetilde{M}$ that reconstructs all functions (among all functions) while the number of **NO** answers remains the same if the input is balanced. The algorithm $\widetilde{M}$ first performs the same queries as M does, until it either has no balanced candidates or a single balanced candidate f compatible with the

answers so far. In the former case, it reconstructs the function by brute-force. In the second case, it performs $\textsc{Query}_1^c(a, f(a)_i, i)$ for all $a \in A$ and $i \in [k]$ to verify that indeed the input is f. If the input is indeed f, we have now distinguished f among all functions (rather than all balanced functions) without additional **NO** answers. If any of the queries answers **NO**, we again have no balanced candidates left and may perform the brute-force approach again.

We will show that, when restricted to balanced functions, $\widetilde{M}$ has an average query complexity (in terms of the number of **NO** answers) greater than $\frac{1}{11}\Delta nk$. Since M has the same number of **NO** answers as $\widetilde{M}$ on balanced inputs, it has the same average query complexity as $\widetilde{M}$. Using Yao's principle (Lemma 2.33), it then follows that any randomised algorithm that reconstructs balanced functions has at least $\frac{1}{11}nk$ queries answered **NO** in expectation.

By Lemma 2.34, there are at most $|B|^n e^{-\frac{1}{50}nk}$ functions $f : A \to B$ for which $\widetilde{M}$ reconstructs f in less than $\frac{1}{10}\Delta nk$ queries. On the other hand the number of balanced function from A to B is the following multinomial coefficient $\binom{n}{c,\dots,c} = \frac{n!}{(c!)^n}$. In particular, there are at least $\binom{n}{c,\dots,c} - (n/c)^n e^{-\frac{1}{50}nk}$ balanced function for which $\widetilde{M}$ requires at least $\frac{1}{10}\Delta nk$ queries. This means that the average query complexity of $\widetilde{M}$ is at least

$$\frac{\binom{n}{c,\dots,c} - (n/c)^n e^{-\frac{1}{50}nk}}{\binom{n}{c,\dots,c}} \frac{1}{10}\Delta nk = \frac{n! - (c!)^n (n/c)^n e^{-\frac{1}{50}nk}}{n!} \frac{1}{10}\Delta nk \geqslant \frac{1}{11}\Delta nk$$

since,

$$(c!)^n (n/c)^n e^{-\frac{1}{50}nk} = \left(\frac{n}{e}\right)^n \left(\frac{ec!}{c} e^{-\frac{1}{50}k}\right)^n \leqslant n^n e^{-\frac{51}{50}n} \leqslant \frac{1}{100} n!$$

using for the first inequality that $k \geqslant 50(c \ln c + 2)$ and for the second that $n \geqslant 2^{51}$. □

Proof of Lemma 2.34. Let R be a deterministic algorithm that uses the coordinate oracle to reconstruct functions. Let F_t denote the set of possible functions $f : A \to B$ that are consistent with the first t queries done by R. (This depends on the input function $g : A \to B$, but we leave this implicit.) For $a \in A$ and $i \in [k]$, let

$$J_{a,i}^t = \{j \in [\Delta] \mid f(a)_i = j \text{ for some } f \in F_t\}.$$

Note that all values $j_1, j_2 \in J_{a,i}^t$ are equally likely in the sense that there is an equal number of $f \in F_t$ with $f(a)_i = j_1$ as with $f(a)_i = j_2$. The algorithm R will perform the same t queries for all $f \in F_t$. In particular, if $g : A \to B$ was chosen uniformly at random, then after the first t queries all $f \in F_t$ are equally likely (as input function) and in particular $g(a)_i$ is uniformly distributed over $J_{a,i}^t$, independently of the sets $J_{a',i'}^t$ for $(a', i') \neq (a, i)$. This is the part for which we crucially depend on the fact that we allow all functions $f : A \to B$ and

not just bijections (where there may be dependencies between the probability distributions of $g(a)$ and $g(a')$ for distinct $a, a' \in A$).

We say that the t^{th} query of the algorithm is *special* if

- it is a Type 1 query $\textsc{Query}_1^c(a, b, i)$ and $|J_{a,i}^t| \geqslant \Delta/2$, or
- it is a Type 2 query $\textsc{Query}_2^c(a, a', i)$ and either $|J_{a,i}^t|$ or $|J_{a',i}^t|$ is at least $\Delta/2$.

Let T denote the number of **NO** answers to *special* queries that R does to the coordinate oracle until it has reconstructed the input function. We let $Y_i = 1$ if the answer of the i^{th} special query is **YES** and 0 otherwise. So $\sum_{i=1}^{T} Y_i$ denotes the number of special queries with answer **YES**.

At the start of the algorithm $J_{a,i}^0 = [\Delta]$ for all $a \in A$ and $i \in [k]$. Thus, to reconstruct the function, the pair (a, i) is either (1) involved in a special query with answer is **YES** or (2) involved in $\Delta/2$ special queries for which the answer is **NO**. Since any query involves at most two elements of A, we deduce that

$$|A|k/2 = nk/2 \leqslant \left(T - \sum_{i=1}^{T} Y_i\right) \frac{2}{\Delta} + \sum_{i=1}^{T} Y_i.$$

We aim to prove that if $g : A \to B$ is sampled uniformly at random, then with high probability $T = T(g) \geqslant \frac{1}{10}\Delta nk$. In order to do so, we consider a simplified process and a random variable τ which is stochastically dominated by T (i.e. for any $x \in \mathbb{R}^+$, $\mathbb{P}(T \leqslant x) \leqslant \mathbb{P}(\tau \leqslant x)$). Let us consider an infinite sequence of i.i.d. random variables $X_1, X_2, X_3, \cdots \sim \text{Bernoulli}(2/\Delta)$. Note that

$$H(t) = \left(t - \sum_{i=1}^{t} X_i\right) \frac{2}{\Delta} + \sum_{i=1}^{t} X_i = \left(1 - \frac{2}{\Delta}\right) \sum_{i=1}^{t} X_i + \frac{2t}{\Delta}$$

is increasing in t. Let τ be the first integer t for which $H(t) \geqslant \frac{1}{2} n \log_\Delta n$.

If g is sampled uniformly at random then the jth special query (say involving $a \in A$ and $i \in [k]$ with $|J_{a,i}^t| \geqslant \Delta/2$) has answer **YES** with probability

$$\mathbb{P}(Y_i = 1) \leqslant \frac{2}{\Delta} = \mathbb{P}(X_i = 1).$$

This is because all values of $J_{a,i}^t$ are equally likely for $g(a)_i$ (and independent of the value of $g(a')_i$ or b_i for $b \in B$ and $a' \in A$). This inequality holds independently of the values of $(Y_1, \ldots, Y_{i-1})$. This implies that, for any $t \in \mathbb{N}^+$ and any $x \in \mathbb{R}^+$,

$$\mathbb{P}\left(\sum_{i=1}^{t} X_i \leqslant x\right) \leqslant \mathbb{P}\left(\sum_{i=1}^{t} Y_i \leqslant x\right).$$

Therefore,

$$\mathbb{P}\left(\left(1-\frac{2}{\Delta}\right)\sum_{i=1}^{t}X_i+\frac{2t}{\Delta}\leqslant x\right)\leqslant\mathbb{P}\left(\left(1-\frac{2}{\Delta}\right)\sum_{i=1}^{t}Y_i+\frac{2t}{\Delta}\leqslant x\right).$$

From this we can conclude that $\mathbb{P}(T\leqslant x)\leqslant\mathbb{P}(\tau\leqslant x)$, thus T stochastically dominates τ.

If $\tau\leqslant\frac{1}{10}\Delta nk$, then using the definition of τ we find that

$$\left(\frac{1}{10}\Delta nk-\sum_{i=1}^{\tau}X_i\right)\frac{2}{\Delta}+\sum_{i=1}^{\tau}X_i\geqslant\frac{1}{2}nk$$

which implies

$$\sum_{i=1}^{\tau}X_i\geqslant\left(1-\frac{2}{\Delta}\right)\sum_{i=1}^{\tau}X_i\geqslant\frac{3}{10}nk.$$

Let $x=\frac{1}{10}\Delta nk$. We compute $\mathbb{E}\left[\sum_{i=1}^{x}X_i\right]=\frac{2}{\Delta}x=\frac{1}{5}nk$. Using Chernoff's inequality (see e.g. [143]) we find

$$\mathbb{P}(\tau\leqslant x)\leqslant\mathbb{P}\left(\sum_{i=0}^{x}X_i\geqslant\left(1+\frac{1}{2}\right)\frac{1}{5}nk\right)\leqslant\exp\left(-\left(\frac{1}{2}\right)^2\frac{1}{5}nk/\left(2+\frac{1}{2}\right)\right).$$

Since $\frac{1}{2}\frac{1}{2}\frac{1}{5}\frac{2}{5}=\frac{1}{50}$, this proves $\mathbb{P}(T\leqslant x)\leqslant\mathbb{P}(\tau\leqslant x)\leqslant e^{-\frac{1}{50}nk}$. In particular, the probability that at most $\frac{1}{10}\Delta k$ queries are used is at most $e^{-\frac{1}{50}nk}$, as desired. □

2.5.2 Reconstructing functions from the word oracle

Let once again $A=[n]$ and $B=[\Delta]^k$. We next turn our attention to reconstructing functions $f:A\to B$ from a more complicated oracle that we use as a stepping stone to get to distance queries in trees. For $b\in B$, we write $b_{[i,j]}=(b_i,b_{i+1},\ldots,b_j)$. It will also be convenient to define $b_\emptyset$ as the empty string. The **word oracle** can answer the following two types of questions.

- **Type 1**. $\textsc{Query}_1^w(a,b)$ for $a\in A$ and $b\in B$, answers the largest $i\in[0,k]$ with $f(a)_{[1,i]}=b_{[1,i]}$.
- **Type 2**. $\textsc{Query}_2^w(a,a')$ for $a,a'\in A$, answers the largest $i\in[0,k]$ with $f(a)_{[1,i]}=f(a')_{[1,i]}$.

By studying the number of queries for the word oracle and the number of **NO** answers for the component oracle, we can link the two reconstruction problems as follows.

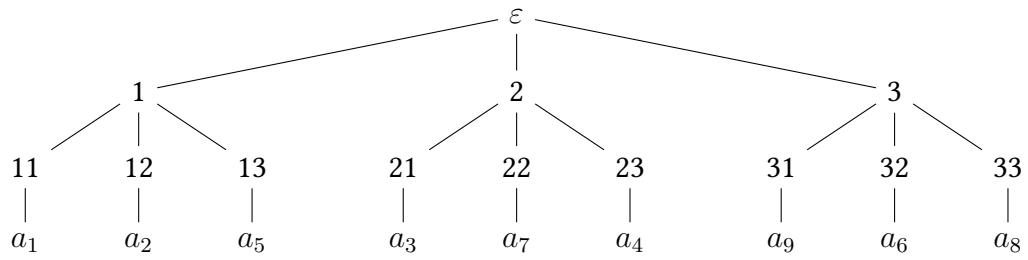


Figure 2.5: Example of the tree $T_{c,\Delta,k}$ constructed in the proof of Theorem 2.14 for $\Delta = 4$, $c = 1$ and $k = 2$ with the labelling ℓ of the internal nodes.

Lemma 2.35. *For all positive integers Δ, k and n, for any algorithm M using the word oracle that reconstructs functions $f : A \to B$ in at most $q(f)$ queries in expectation, there exists an algorithm M' using the coordinate oracle that reconstructs functions $f : [n] \to [\Delta]^k$ such that at most $q(f)$ queries are answered **NO** in expectation.*

Proof. Given an algorithm M using the word oracle, we build a new algorithm M' using the coordinate oracle. We do so query-by-query. If M asks $\textsc{Query}_1^w(a, b)$, then M' performs a sequence of queries $\textsc{Query}_1^c(a, b_1, 1)$, $\textsc{Query}_1^c(a, b_2, 2)$, $\ldots$, $\textsc{Query}_1^c(a, b_{i+1}, i+1)$, where $i \in [0, k-1]$ is the largest for which $f(a)_{[1,i]} = b_{[1,i]}$. Note that the sequence indeed simulates a query of the word oracle yet the coordinate oracle answers **NO** at most once (on the $(i+1)$th query).

Queries of Type 2 can be converted analogously. This way, for every input f, the natural "coupling" of the randomness in M and M' ensures that the number of **NO** answers to M' is stochastically dominated by the number of queries to M. In particular, the expected number of **NO** answers given by M' is upper bounded by $q(f)$, the expected number of queries to M. □

Lemma 2.35 and Theorem 2.32 now give the following result.

Corollary 2.36. *Let $\Delta \geqslant 3$, $c \leqslant \Delta - 1$ and $k \geqslant 50(c \ln c + 2)$ be positive integers. Let $n = c\Delta^k$. Any algorithm reconstructing $f : [n] \to [\Delta]^k$ using the word oracle, in the special case where f is known to be balanced, needs at least $\frac{1}{11}\Delta nk$ queries in expectation.*

2.5.3 Reducing tree reconstruction to function reconstruction

In order to prove Theorem 2.14 we consider a specific tree $T_{c,\Delta,k}$ (with $c \leqslant \Delta$): the tree of depth $k+1$ where each node at depth at most $k-1$ has exactly Δ children and each node at depth k has exactly c children (see Fig. 2.5). Theorem 2.14 is an almost direct consequence of the following lemma.

Lemma 2.37. *Let $\Delta \geqslant 2$, $k \geqslant 150$ be positive integers. Consider a labelling of the tree $T_{\Delta,k}$ with $N = \Delta^k$ leaves, where only the labels of the leaves are unknown. Any randomised algorithm requires at least $\frac{1}{11}\Delta N \log_\Delta N$ queries in expectation to reconstruct the labelling.*

Proof. We consider $T = T_{c,\Delta,k}$ and let L be the set of leaves of T and let P be the set parents of the leaves. The tree T has $n = \sum_{i=0}^{k} \Delta^i + c\Delta^k$ nodes and $N = c\Delta^k$ leaves. Let $p : L \to P$ be the bijection that sends each leaf to its direct parent. We label internal nodes as follows. The root is labelled $\emptyset$ (the empty string) and if a node v has label ℓ and has Δ children, then we order the children $1, \ldots, \Delta$ and we label the child i with label obtained from concatenation $\ell + (i)$. We put such labels on all internal nodes.

Let I denote the set of internal nodes and let $\ell(v)$ denote the label of $v \in I$. Let $f : L \to [\Delta]^k$ be the bijection that sends a leaf $u \in L$ to the label $\ell(p(u)) \in [\Delta]^k$ of its direct parent.

We consider the trees which have a fixed labelling (as described above) for node in I, and every possible permutation of the labelling of the leafs. All possible bijections $f : L \to [\Delta]^k$ appear among the trees that we are considering. To reconstruct the tree, we in particular recover the corresponding bijection f. Distance queries between internal vertices always give the same response and can be ignored. We show the other queries are Type 1 and Type 2 queries in disguise.

- For $a \in L$ and $b \in I$ the distance between a and b is given as follows. Let $z \in I$ be the nearest common ancestor of a and b and say z has depth i and b has depth j. The distance between a and b is $1 + (k - i) + (j - i)$. The values of k and j do not depend on f but the value of i is exactly given by $\max\{s : f(a)_{[0,s]} = b_{[0,s]}\}$, the answer to the corresponding type 1 query of (a, b) to the word oracle. To be precise, since b may have a length shorter than i, the query $\textsc{Query}_1^w(a, b')$ where $b'_s = b_s$ for all $s \in [1, |b|]$ and $b'_s = 0$ otherwise, gives the desired information.

- For $a, a' \in L$, the distance between a and a' is given by $2(1 + (k - i))$ for i the answer of a Type 2 $\textsc{Query}_2^w(a, a')$ to the word oracle.

This shows that we reduce an algorithm to reconstruct the labelling of the leaves from q distance queries to an algorithm that reconstructs functions $f : L \to [\Delta]^k$ from q queries to the word oracle. By Theorem 2.32, since $k \geqslant 50(c \ln c + 2)$, we need at least $\frac{1}{11}\Delta N k$ queries. □

We are now ready to deduce the main result of this section.

Theorem 2.14. *Let $\Delta \geqslant 2$ and $n = 2c\Delta^k$ be integers, where $c \in [1, \Delta)$ and $k \geqslant 50(c \ln c + 3)$ is an integer. Any randomised algorithm requires at least $\frac{1}{50}\Delta n \log_\Delta n$ queries in expectation to reconstruct n-vertex trees of maximum degree $\Delta + 1$.*

Proof. Let $\Delta, n \geqslant 2$ be integers. We write $n = 2c\Delta^k$ for $c \in [1, \Delta)$ and k an integer. (When $n/2 \geqslant 1$, there is a unique pair $(c, k) \in [1, \Delta) \times \mathbb{Z}_{\geqslant 0}$ with $n/2 = c\Delta^k$.)

Suppose that $k \geqslant 50(c \ln c + 3)$. In particular, $\Delta \geqslant 2$ implies $k \geqslant \lfloor \log_\Delta n - 1 \rfloor$. The tree $T = T_{\lfloor c \rfloor, \Delta, k}$ considered in Lemma 2.37 has maximum degree $\Delta + 1$, $N = \lfloor c \rfloor \Delta^k$ leaves and $n' = \sum_{i=0}^{k} \Delta^i + \lfloor c \rfloor \Delta^k$ vertices, where

$$n/4 \leqslant N \leqslant n' \leqslant 2c\Delta^k = n.$$

For $\Delta \geqslant 2$ and $c \geqslant 1$, if $n \geqslant 2\Delta^{50(c \ln c + 3)}$ then $N \geqslant \frac{n}{4} \geqslant \Delta^{50(c \ln c + 2)}$. So we may apply Lemma 2.37 and find that at least

$$\frac{1}{11} \Delta N \lfloor \log_\Delta N - 1 \rfloor \geqslant \frac{1}{44} \Delta n (\log_\Delta n - 4)$$

queries are required. As $\log_\Delta n \geqslant 150$, we find that this is at least $\frac{1}{50} \Delta n \log_\Delta n$. □

Corollary 2.38. *Any randomised algorithm requires at least $\frac{1}{50} \Delta n \log_\Delta n$ distance queries to reconstruct n-vertex trees of maximum degree $\Delta + 1 \geqslant 3$ if $n \geqslant 2\Delta^{50(\Delta \ln \Delta + 3)}$.*

2.5.4 Randomised lower bounds for related models

We next show that our result implies various other new randomised lower bounds. Although we state these results with a weaker assumption on n for readability reasons, we remark that our more precise set-up (allowing Δ to be a small polynomial in n for specific values of n) also applies here.

Betweenness queries A *betweenness query* answers for three vertices (u, v, w) whether v lies on a shortest path between u and w. Using three distance queries to (u, w), (u, v) and (v, w), you can determine whether v lies on a shortest path between u and w, so the betweenness oracle is weaker (up to multiplicative constants). It has been shown in [2] that randomised algorithms can obtain a similar query complexity for betweenness queries as was obtained for distance queries by [107]. Moreover, a randomised algorithm for 4-chordal graphs has been given that uses a quasi-linear number of queries to a betweenness oracle [162]. A deterministic algorithm using $\tilde{O}(\Delta n^{3/2})$ betweenness queries has been given for trees, as well as a $\Omega(\Delta n)$ lower bound [97]. Our randomised lower bound from Theorem 2.14 immediately extends to this setting.

Corollary 2.39. *Any randomised algorithm requires at least $\frac{1}{150} \Delta n \log_\Delta n$ betweenness queries to reconstruct n-vertex trees of maximum degree $\Delta + 1 \geqslant 3$ if $n \geqslant 2\Delta^{50(\Delta \ln \Delta + 3)}$.*

Path and comparison queries Given two nodes i, j in a directed tree, a *path query* answers whether there exists a directed path from i to j. Improving on work from [181], it was shown in [5] that any algorithm needs $\Omega(n \log n + n\Delta)$ to reconstruct a directed tree on n nodes of maximum degree Δ. When we consider a directed rooted tree in which all edges are directed from parent to child, then path queries are the same as *ancestor queries*: given u, v in a rooted tree, is u an ancestor of v? We apply this to the tree $T_{c,\Delta,h}$ from Lemma 2.37 for which the labels of all internal vertices are fixed but the labels of the leaves are unknown. Path queries (u, v) only give new information if v is a leaf and u is an internal vertex. But this is weaker than distance queries, since we can obtain the same information by asking the distance between u and v. This means that we can redo the calculation from the proof of Theorem 2.14 (applying Lemma 2.37) to lift the lower bound to path queries.

Corollary 2.40. *Any randomised algorithm requires at least* $\frac{1}{50}\Delta n \log_\Delta n$ *path queries to reconstruct* n*-vertex directed trees of maximum degree* $\Delta + 1 \geqslant 3$ *if* $n \geqslant 2\Delta^{50(\Delta \ln \Delta + 3)}$.

A randomised algorithm using $O(n \log n)$ path queries on bounded-degree n-vertex trees has been given in [5] but their dependency on Δ does not seem to match our lower bound. We remark that besides query complexity, works on path queries such as [5, 4, 181] also studied the round complexity (i.e. the number of round needs when queries are performed in parallel).

The same ideas applies to lift our lower bound to one for reconstructing tree posets $(T, >)$ from *comparison queries*, which answer for given vertices u, v of the tree whether $u < v, v < u$ or $u||v$.

Corollary 2.41. *Any randomised algorithm requires at least* $\frac{1}{50}\Delta n \log_\Delta n$ *comparison queries to reconstruct* n*-vertex tree posets of maximum degree* $\Delta + 1 \geqslant 3$ *if* $n \geqslant 2\Delta^{50(\Delta \ln \Delta + 3)}$.

This improves on the lower bound of $\Omega(\Delta n + n \log n)$ from [164] and matches (up to a $(C \log \Delta)$ factor) the query complexity of their randomised algorithm.

Membership queries for reconstructing partitions The (n, k)-partition problem was introduced by King, Zhang and Zhou [117]. Given n elements which are partitioned into k equal-sized classes, the partition needs to be determined via queries of the form "Are elements a and b in the same class?". They used the adversary method to prove $\Omega(nk)$ queries are needed by any deterministic algorithm. Liu and Mukherjee [131] studied this problem phrased as learning the components of a graph via membership queries (which answer whether given vertices lie in the same component or not) and provide an exact deterministic lower bound of $(k-1)n - \binom{k}{2}$ for deterministic algorithms. It indeed seems natural that randomised algorithms need αkn queries for some constant α in this setting.

Nonetheless the best lower bound for randomised algorithm seems to be the information-theoretic lower bound of $\Omega(n \log k)$. Our next result remedies this gap in the literature.

Corollary 2.42. *Let $\varepsilon > 0$. Any randomised algorithm requires at least $\frac{1}{11}nk$ membership queries to solve the (n,k)-partition problem if $n \geqslant k^{1+\varepsilon}$ is sufficiently large.*

Proof. Since we plan to apply Lemma 2.34, we will write $\Delta = k$.

First note that we can see the (n,Δ)-partition problem using *membership* queries as reconstructing a balanced function using only Type 2 queries to the coordinate oracle. Formally, if $f : [n] \to [\Delta]$ is the function which associates an element $a \in [n]$ to the index $i \in [\Delta]$ of the part that contains a (out of the Δ parts in the partition), then a membership query between $a, a' \in [n]$ is exactly equivalent to the coordinate query $\textsc{Query}_2^c(a,a')$ applied the function f. Once we reconstructed the partition, we can retrieve the index of each parts using $\Delta^2 = o(n\Delta)$ queries of Type 1 to the coordinate oracle. Therefore it suffices to show that at least $\frac{1}{11}\Delta n$ queries are needed in expectation to reconstruct a balanced function using the coordinate oracle.

Applying Lemma 2.34 with $k = 1$, we find that when f is sampled uniformly at random (among all functions $g : [n] \to [\Delta]$), the probability that a given randomised algorithm uses less than $\frac{1}{10}n\Delta$ queries is at most $e^{-\frac{1}{50}n}$. In particular, the number of balanced functions reconstructed in less than $\frac{1}{10}n\Delta$ queries is upper bounded by $\Delta^n e^{-\frac{1}{50}n}$. We compare this number to the total number $\binom{n}{n/\Delta,\dots,n/\Delta} = \frac{n!}{(n/\Delta)!^\Delta}$ of balanced functions:

$$\begin{aligned}\frac{\Delta^n e^{-\frac{1}{50}n}}{\frac{n!}{(n/\Delta)!^\Delta}} &\leqslant \Delta^n e^{-\frac{1}{50}n} \frac{(2\pi n/\Delta)^{\Delta/2}(n/(e\Delta)^n}{\sqrt{2\pi n}(n/e)^n} e^{\Delta/(12n/\Delta)} \\ &= \frac{1}{\sqrt{2\pi n}}(2\pi n/\Delta)^{\Delta/2} \exp(\Delta^2/(12n) - n/50).\end{aligned}$$

Here we used that for all $n \geqslant 1$,

$$\sqrt{2\pi n}(n/e)^n < n! < \sqrt{2\pi n}(n/e)^n e^{1/(12n)}.$$

Since $\Delta \leqslant n^{1/(1+\epsilon)}$ for some $\epsilon > 0$, the fraction tends to 0, so in particular becomes smaller than $\frac{1}{100}$ when n is sufficiently large (depending on ϵ). This implies that the expected number of queries to reconstruct a Δ-balanced function is at least $\frac{99}{100}\frac{1}{10}\Delta n \geqslant \frac{1}{11}n\Delta$. By the discussion at the start, we find the same lower bound for the (n,Δ)-partition problem. □

The same lower bound holds when queries of the form "Is element a in class i?" are also allowed. We expect that our methods can be adapted to handle parts of different sizes and that our constant $\frac{1}{11}$ can be easily improved.

Randomised algorithms have been studied in a similar setting by Lutz, De Panafieu, Scott and Stein [133] under the name *active clustering*. They provide the optimal average query complexity when the partition is chosen uniformly at random among all partitions. They also study the setting in which a partition of n items into k parts is chosen uniformly at random and allow queries of the form "Are items i and j in the same part?". However there is a key difference: the set of answers the algorithm receives, needs to distinguish the partition from any other partition (including those with a larger number of parts). In this setting, the following algorithm is optimal. Order the items $1, \dots, n$. For $i = 1, \dots, n$, query item i to items $j = 1, \dots, i$ in turn if the answer to the query "Are items i and j in the same part?" is not yet known. It follows from [133, Lemma 9] that this algorithm is has the lowest possible expected number of queries. The expected number of queries used is at most

$$(1 + 2 + 3 + \cdots + k)\frac{1}{k}n = \frac{k+1}{2}n.$$

In particular, for $i = \Omega(\log n)$ queries, with high probability there are items in k different parts among the first $i - 1$ items. Since the number of parts is not "known", the algorithm will use k queries for item i if is in the "last part" and so the complexity is $(\frac{k+1}{2} + o(1))n$ as $n \to \infty$. The same algorithm would use $(\frac{k+1}{2} - \frac{1}{k} + o(1))n$ queries when the number of parts may be assumed to be at most k, in which case queries "to the last part" are never needed. So the assumption on whether the number of parts is known, changes the query complexity. Together with the parts "being known to be exactly balanced", this introduces additional dependencies in the (n, k)-partition problem that our analysis had to deal with.

Phylogenetic reconstruction This setting comes from biology. Reconstructing a phylogenetic tree has been modelled via what we call *leaf-distance* queries (similarity of DNA) between leaves of the input tree [90, 182, 117]. Although very similar, the query complexities of the phylogenetic model and the distance query model on trees are not directly related. In the phylogenetic model, the set of leaves is already known and the leaf-distance queries are only possible between leaves. Moreover, we consider a phylogenetic tree to be reconstructed once we know all the pairwise distances between the leaves. For example, if the input tree is a path on n vertices, then in the phylogenetic setting we receive only two leaves and are finished once we query their distance, whereas in the distance reconstruction setting it takes $\Omega(n)$ queries to determine the exact edge set.

Improving on various previous works [42, 106, 110], King, Zhang, and Zhou [117, Theorem 3.2] showed that any deterministic algorithm reconstructing phylogenetic trees of maximum degree Δ with N leaves needs at least $\Omega(\Delta N \log_\Delta N)$ leaf-distance queries. Deterministic algorithms achieving this complexity are also known [90].

For randomised algorithms, the previous best lower bound was the information-theoretic

lower bound of $\Omega(N \log N / \log\log N)$. We provide the following randomised lower bound from Lemma 2.37 which is tight up to a multiplicative constant.

Theorem 2.43. *Let $\Delta \geqslant 2$, $c \leqslant \Delta - 1$ and $k \geqslant 50(c \ln c + 2)$ be positive integers. Let $N = c\Delta^k$. Any randomised algorithm reconstructing phylogenetic trees of maximum degree $\Delta + 1$ with N leaves needs at least $\frac{1}{20}\Delta N \log_\Delta N$ leaf-distance queries in expectation.*

Proof. Let $T = T_{c,\Delta,k}$ be the tree considered in Lemma 2.37 with N leaves.

Suppose towards a contradiction that we could obtain the pairwise distances between the leaves of this tree in $\frac{1}{20}\Delta N \log_\Delta N$ leaf-distance queries in expectation. We show that, from this, we can recover the labels of the leaves of T using only $\Delta^2 N \leqslant \frac{1}{30}\Delta N \log_\Delta N$ additional distance queries, contradicting Lemma 2.37 since $\frac{1}{20} + \frac{1}{50} \leqslant \frac{1}{11}$ and $\log_\Delta N \geqslant k \geqslant 50$.

We proceed by induction on k, the depth of the tree T. When $k = 0$, $T_{c,\Delta,0}$ is a star with c leaves. There is nothing to prove, as the parent of each leaves in known to be the root. Suppose $k \geqslant 1$ and that the claim has already been shown for smaller values of k. We define an equivalence relation on the set of leaves: for $u_1, u_2 \in L$, $u_1 \sim u_2$ if and only if $d(u_1, u_2) < 2k$. This is an equivalence relation with Δ equivalence classes, as $d(u_1, u_2) < 2k$ if and only if u_1 and u_2 have a child of the root as common ancestor.

Let $u_1, u_2, \ldots, u_\Delta$ be arbitrary representatives of each of the Δ equivalence classes. (Note that we can select these since we already know the distances between the leaves.) Let r denote the root of T. We ask $\textsc{Query}(u_i, N(r))$ for all $i \in [\Delta]$. From this we can deduce the common ancestor among the children of the root for each of the classes. It is the unique neighbour of r lying on a shortest path from u_i to r. Let V_i denote the set of all the leaves that have the i^{th} neighbour of r as common ancestor. We also define T_i to be the subtree rooted in the i^{th} neighbour of r. We remark that it is now sufficient to solve Δ subproblems of reconstructing T_i knowing each V_i leaf matrix. By the induction hypothesis, each subproblem is solvable in $|V(T_i)|\Delta^2 = \frac{N-1}{\Delta}\Delta^2 = (N-1)\Delta$ queries. Therefore, in total this algorithm uses $(N-1)\Delta^2 + \Delta^2 = \Delta^2 N$ distance queries. □

2.5.5 Lower bound for reconstructing a Voronoi cell locally

One of the two main steps of the algorithm designed by Kannan, Mathieu, and Zhou in Theorem 2.3 is based on reconstructing Voronoi cells locally. In this section, we prove the following theorem. Intuitively it states that the brute-force subalgorithm used in this step is optimal.

Theorem 2.7. *For any $n \in \mathbb{N}$, there exists a family of graphs $\mathcal{G}$, all on the same vertex set $V := X \sqcup Y$, and there exists $x \in V$ and $S \subseteq V$ satisfying the following properties, for any $G, G' \in \mathcal{G}$:*

- *G has maximum degree 5,*
- *$X = \mathrm{Vor}_G(x, S)$,*
- *$G[Y] = G'[Y]$,*
- *$|V| = \Theta(n^2)$ and $|X| = \Theta(n \log n)$.*

Moreover, any randomised reconstruction algorithm using the partial oracle $\textsc{Query}_{G|X}$, taking as input a graph $G \in \mathcal{G}$, uses $\Omega(n^2)$ queries to uniquely reconstruct $G[X]$ in expectation.

Before starting the proof, let us define *starshaped* sets, already introduced in the context of reconstruction by Abrahamsen, Bodwin, Rotenberg and Stockel [2].

Definition 2.44. *Given a graph G, a vertex set $X \subseteq V(G)$ is starshaped with respect to $x \in X$ if, for all $v \in X$, every shortest path in G from x to v is contained in X.*

Note that by definition a Voronoi cell is starshaped. Being starshaped is also a sufficient condition for being a Voronoi cell in the following sense. For any starshaped set X in some graph G, there exists a graph H and a set of centers S such that for some $s \in S$, there is an isometry between X in G and $\mathrm{Vor}_H(s, S)$ in H. We are now equipped to prove Theorem 2.7.

Proof of Theorem 2.7. Let us start with the following claim, useful for the construction of $\mathcal{G}$.

Claim 2.45. *For any $k \in \mathbb{N}$, there exists a graph H of maximum degree 3, and a set $K \subseteq H$ with $|K| = k$, and $|V(H)| \leqslant 4k^2$ such that,*

$$\forall u, v \in K, u \neq v \implies d_H(u, v) = 2\lceil \log_2 k \rceil.$$

Proof. Let us consider the graph H, on vertex set $K \cup L$. H consists of k disjoint complete binary trees $(H_u)_{u \in K}$, where, for every $u \in K$, H_u is rooted at u and has depth $\lceil \log_2 k \rceil$. Additionally, consider k arbitrary leaves of H_u and denote them by $\ell_u(v)$ for $v \in K$. To obtain H we add, for every distinct $u, v \in B$, an edge from $\ell_u(v)$ to $\ell_v(u)$. As claimed, for any distinct $u, v \in K$, $d(u, v) = 2\lceil \log_2 k \rceil$. Moreover, the size of $V(H)$ is at most $k2^{\log_2 k + 2} = 4k^2$. □

We now construct a family of graphs $\mathcal{G}$ on vertex set V, such that for some $X \subseteq V$, X is starshaped in every graph in $\mathcal{G}$ for some vertex $x \in X$. We construct $\mathcal{G}$ on the model pictured in Fig. 2.6.

Without loss of generality, suppose that n is a power of 2 (otherwise consider the largest power of 2 smaller than n). Let $T_{m,\ell}$ denote the tree obtained by subdividing ℓ times every

edge connected to a leaf in a binary tree with m leaves. Let us define the "model" graph G_0 from which every graph in $G \in \mathcal{G}$ will be built. Let $G_0[X]$ be a copy of $T_{n,10\log_2(n)-1}$ rooted in some arbitrary vertex $x \in X$. Let $A = \{a \in X \mid d(x,a) = \log_2 n\}$ and let B denote the set of all leaves in $G_0[X]$. let $C = \{u \in X \mid d(x,u) \leqslant \log_2 n\}$ be the set of vertices "above" A.

We construct $G_0[V \setminus X]$ by taking two copies of $T_{n,4\log_2 n}$ rooted in the same vertex y and identifying the leaves of the first copy with the vertices of A and the leaves of the second copy with the vertices of B. By doing so, we ensure that, through $G_0[V \setminus X]$, the distance for any pair $(a,b) \in A \times B$ is at most $2(4\log_2 n + \log_2 n) = 10\log_2 n$.

Finally, to obtain G_0, we add a copy of the graph H satisfying Claim 2.45, where we identify K and B.

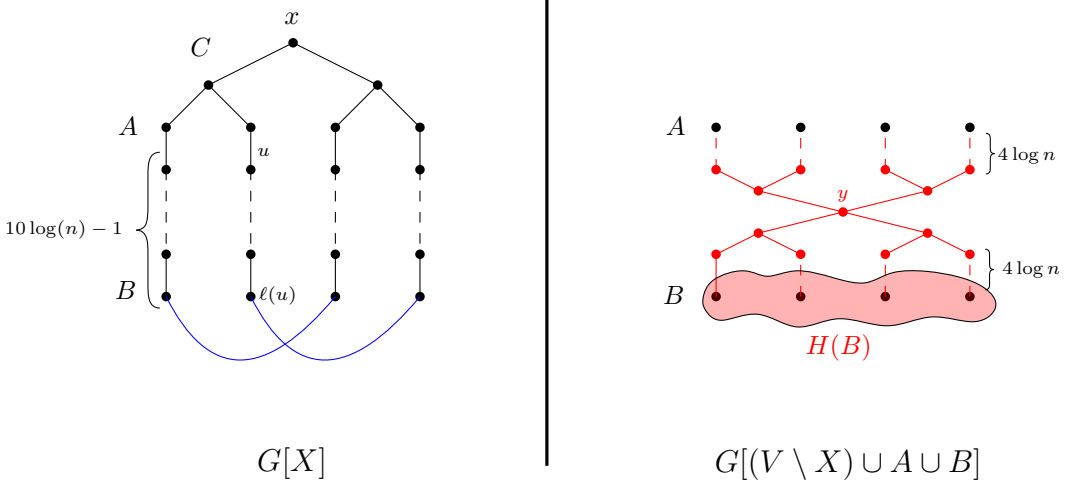


Figure 2.6: Depiction of a graph G in the class $\mathcal{G}$ constructed in the proof of Theorem 2.7. On the left only the starshaped set $G[X]$ is depicted and on the right the rest of the graph is represented. The edges in blue denote the edges of M.

For every matching M over B, we add a copy of $G_M = (V, E(G_0) \cup M)$ to $\mathcal{G}$. See Fig. 2.6 for a representation. Note that, for any $G \in \mathcal{G}$, X is indeed starshaped in G with x as center. Moreover, it is possible to add a subdivided binary tree where the leaves are identified with $N(A) \setminus X$ such that the root r of this new tree is at distance exactly $10\log_2 n$ from $N(A) \setminus X$. By doing so, we ensure that $X = \mathrm{Vor}_G(x, \{x, r\})$ for every $G \in \mathcal{G}$.

The remaining of the proof aims to show that any single query to the oracle $\textsc{Query}_{G|X}$ over $\mathcal{G}$ can be simulated using a single query from the simpler oracle $\textsc{Match}$ that takes as input two vertices $b, b' \in B$ and answer YES if $bb' \in M$ and NO otherwise.

For any vertex $u \in X \setminus C$, let $\ell(u)$ denote the endpoint in B of the path in $X \setminus C$ containing u. Let us first show two key properties of our construction.

Claim 2.46. *For any $G \in \mathcal{G}$ and any pair $(u, x) \in C \times X$ we have $d_G(u, x) = d_{G_0}(u, x)$.*

Proof. Consider $(u, x) \in C \times X$ and consider a shortest path P from u to x that minimises the number of edges in M used by P. We will show that $P \cap M = \emptyset$ from which the claim follows directly. Note that P contains a vertex $a \in A$ (potentially $a = u$). Suppose towards a contradiction that it contains an edge in $b_1b_2 \in M$, therefore P can be written as $P = u \dots a \dots b_1b_2 \dots v$. By construction $d_G(a, b_1) \geqslant 10 \log_2(n) - 1$, therefore we have $d(a, b_2) \geqslant 10 \log_2 n$. It suffices now to consider the path P' obtained from P by substituting the subpath between a and b_2 by the path between a and b_2 in $G_0[V \setminus X]$, as mentioned above, this path as length exactly $10 \log_2 n$, therefore P' is also a shortest path and contains one less edge in M contradicting the minimality of P. □

Claim 2.47. *For any $G \in \mathcal{G}$, and any $u, v \in X \setminus C$ there exists a shortest path P from u to v such that $E(P) \cap M \subseteq \{\ell(u)\ell(v)\}$.*

Proof. Suppose towards a contradiction, that the statement does not hold. Consider $u, v \in X$ a counter-example to the statement minimising $d_G(u, v)$. There exists a shortest path P of the form $P = u \dots b_1b_2 \dots v$ where $b_1b_2 \in M$ and $b_2 \notin \{\ell(u), \ell(v)\}$. If P contains a vertex $w \in C$, using Claim 2.46 on the subpath of P from u to w and from w to v, we obtain a shortest path P' satisfying the claim, therefore we can assume $P \cap C = \emptyset$. The minimality condition allows us to assume $u = b_1$, and since $P \cap C = \emptyset$, $\ell(v)$ is a cut-vertex separating the component of $u = b_1$ and v, therefore P can be written as $P = b_1b_2 \dots \ell(v) \dots v$. Since, by assumption, $b_2 \neq \ell(v)$ and $b_2\ell(v) \notin M$, any shortest path from b_2 to $\ell(v)$ goes through H, and thus the property of H ensures by Claim 2.45 implies that $d_G(b_2, \ell(v)) = 2 \log n$. On the other end, Claim 2.45 also implies that $d_G(b_1, \ell(v)) \leqslant 2 \log_2 n$, but we just showed that the subpath $b_1b_2 \dots \ell(v)$ is of length exactly $2 \log n + 1$ which is a contradiction as any subpath of a shortest path is also a shortest path. □

Suppose w.l.o.g. before making its first query, the algorithm reconstructing G already knows every edge in G except for the edges of M, then Claim 2.46 and Claim 2.47 together imply that the answer to any query $\textsc{Query}_{G|X}(u, v)$ can differ from $d_{G_0}(u, v)$ only if $\ell(u)\ell(v) \in M$. Therefore, $d_G(u, v)$ can be recovered from the query $\textsc{Match}(\ell(u), \ell(v))$. This implies that the number of queries required to reconstruct $G \in \mathcal{G}$ using $\textsc{Query}_{G|X}$ is at least the number of queries needed to reconstruct a matching M on n-vertice, among all possible matchings, using the oracle $\textsc{Match}$. This remark, combined with Claim 2.48 below allows us to conclude that the average complexity of reconstructing $\mathcal{G}$ using $\textsc{Query}_{G|x}$ is at least $n^2/36$.

Claim 2.48. *Any randomised algorithm using the oracle* MATCH *to reconstruct M among the set of matching on n vertices uses at least $n^2/36$ queries in expectation.*

Proof. Let Q and R be disjoint vertex sets with $2|Q| = |R| = \frac{2}{3}n$. We apply Yao's Minimax Principle (Lemma 2.33) to analyse the complexity of reconstructing a random bipartite matching M between Q and R that saturates Q, where M is sampled uniformly at random. It remains to prove that any deterministic algorithm A must make at least $n^2/36$ queries in expectation to reconstruct such a matching M drawn from the above distribution.

Fix this algorithm A and denote by C the expected query complexity of A. For every $q \in Q$, let I_q be the random variable associated to the number of queries made by A involving the vertex q. The linearity of the expectation implies,

$$C = \sum_{q \in Q} \mathbb{E}[I_q].$$

Let us upper bound $\mathbb{E}[I_q]$ for $q \in Q$. We assume w.l.o.g. that every other query of the form $\textsc{Match}(q', r)$, for $q' \in Q \setminus q$ and $r \in R$, has already been asked by the algorithm. Therefore, every edge of M except the one containing q is already known by A. Since $|R| - |Q| \geqslant n/3$, there exists a set $R' \subseteq R$ of size at least $n/3$ such that no element of R' is adjacent to any element of $Q \setminus q$. Moreover, every $r' \in R'$ has the same probability of being the neighbour of q, since M was sampled uniformly at random. This implies, using Markov's inequality, that $\mathbb{E}[I_q] \geqslant (n/6)\mathbb{P}(I_q \geqslant n/6) \geqslant n/12$, and therefore

$$C = \sum_{q \in Q} \mathbb{E}[I_q] \geqslant \sum_{q \in Q} n/12 = n^2/36,$$

which concludes the proof. □

□

2.6 Conclusion and perspectives

The main open question in distance reconstruction, stated by Mathieu and Zhou [140] remains open to this day.

Conjecture 2.5. *For any $\Delta \in \mathbb{N}$ there exists an algorithm A that reconstructs the class of all connected n-vertex graphs of maximum degree Δ using $O_\Delta(n \operatorname{polylog} n)$ queries w.h.p..*

In the light of Section 2.3 and Theorem 2.14, where we established that a query complexity of $f(\Delta)n \log n$ is tight (up to the function f) for large classes of graphs, we ask whether the following strengthening of Mathieu and Zhou's conjecture holds.

Conjecture 2.49. *For any $\Delta \in \mathbb{N}$ there exists an algorithm A which reconstructs the class of all n-vertex graphs of maximum degree Δ using $O_\Delta(n \log n)$ queries w.h.p..*

In pursuit of Conjecture 2.5, researchers have studied the query complexity of bounded-degree graphs under additional constraints. The aim is to progressively weaken these constraints while maintaining a query complexity of $n \operatorname{polylog}(n)$. The results presented in Section 2.3 follow this approach. In particular, we proved in Theorem 2.11 that well-structured tree decompositions can be leveraged in an algorithm. Despite our effort, our approach still requires the bags of the tree decomposition to be "local". Taking a step further towards Conjecture 2.5 could involve tackling the following problem, where the "local" condition is lifted.

Problem 2.50. *What is the query complexity of reconstructing the class of graphs of bounded degree and bounded treewidth?*

Towards Conjecture 6.1, a more fine-grained analysis is required. For example, in Theorem 2.9, we designed an algorithm for k-chordal graphs. We did not try to optimise the dependency on the maximum degree Δ and k and the dependency on k can probably be improved upon. If Conjecture 2.5 holds, then the dependency on k should be at most polylogarithmic. It would be interesting to see if any dependency on k is needed. To ensure that such a new lower bound exploits the chordality, we pose the following problem.

Problem 2.51. *Is it true that for some fixed values of k and Δ and for all n sufficiently large, any algorithm reconstructing k-chordal graphs on n vertices of maximum degree Δ requires at least $10^6 \Delta n \log_\Delta n$ queries?*

For deterministic algorithms determining the correct constant could be achievable, yet would require new insight. In particular, our simple algorithm (Theorem 2.10 may be optimal up to an additive constant. We pose the following problem towards this.

Problem 2.52. *Is there a constant $c > \frac{1}{2}$ such that for all $\Delta \geqslant 3$, there are infinitely many values of n for which any deterministic algorithm that reconstructs n-vertex trees of maximum degree Δ needs at least $c\Delta n \log_\Delta n$ queries?*

We showed in this chapter that the randomised and deterministic query complexities have the same dependence on n and Δ (Theorem 2.10 and Theorem 2.14). In that sense randomness does not help much. Nevertheless, a simple randomisation trick allowed us to gain a factor 2 between Theorem 2.10 and Theorem 2.20. If the answer to the question above is positive, it would show that randomness does at least make a difference.

Finally, we would like to discuss a question that has both a theoretical and practical interest. In Section 2.5 we studied the query complexity of reconstructing $G(n, p)$ for small values

of p. Our result, similarly to [141, 126], heavily relies on the expander structures of random graphs and random regular graphs, and the tame distribution of the degree sequences in these models. In the context of reconstructing Internet networks, studying classes of graphs with less concentrated degree sequences would be natural. In particular, experience shows that real-life networks tend to have a power-law degree distribution, therefore we ask the following question.

Problem 2.53. *Given a degree sequence D following a power-law distribution, what is the complexity of reconstructing a uniformly sampled graph with degree sequence D?*

Chapter 3

Partial ordered set saturation and parameters

This chapter explores topics related to partially ordered set saturation and includes the study of new structural poset parameters that naturally arose during this work. The results presented about saturation are jointly published works with Carla Groenland and Tom Johnston, one co-authoring Hugo Jacob [28] and the other Maria-Romina Ivan [27]. The results about structural poset parameters are joint work with Jędrzej Hodor, Hoang La and William T. Trotter.

3.1 Introduction

Given a graph H, we say that another graph G is H-saturated if G does not contain H as a subgraph, but adding any edge to G creates a copy of H. A fundamental problem in extremal graph theory is to compute the minimum and maximum number of edges of H-saturated graphs. When H is the complete graph on k vertices, two highly celebrated theorems — one by Turán [178] and the other by Erdős, Hajnal, and Moon [68] — solve this question for the maximum and the minimum number of edges, respectively.

Theorem 3.1 ([178]). *For any fix integer $r \geqslant 1$, the maximal number of edges in a K_r-saturated graph on n vertices denoted $\mathrm{ex}(n, K_r)$ satisfies*

$$\mathrm{ex}(n, K_r) = \left(1 - \frac{1}{r}\right) \frac{n^2}{2}.$$

Theorem 3.2 ([68]). *For any fixed integer $r \geqslant 1$, the minimal number of edges in a K_r-saturated graph on n vertices is*

$$n(r-2) - \binom{r-1}{2}.$$

In recent years, there has been significant interest in Turán-type problems for *partially ordered sets*, also called *posets*. We only consider finite posets. This chapter focuses on a poset version of the Erdős-Hajnal-Moon theorem called *induced saturation*. The hypercube, also called the Boolean lattice, is the poset induced by the subsets of $\{1, \dots, n\}$ ordered by the inclusion relation; we denote it by $2^{[n]}$.

Definition 3.3. *Given a poset P, a subset poset $\mathcal{F} \subseteq 2^{[n]}$ is said to be P-saturated if both following conditions are satisfied:*

1. *$\mathcal{F}$ does not contain any induced copy of P;*
2. *for all $x \in Q_n \setminus \mathcal{F}$, $\mathcal{F} \cup \{x\}$ contains an induced copy of P.*

We also define $\mathrm{sat}^*(n, P)$ as the minimum size of a P-saturated family in $2^{[n]}$.

The parameter sat^* was introduced first by Gerbner, Keszgeh, Lemons, Palmer, Pálvölgyi and Patkós [82]. Since then there has been extensive effort to calculate its value and asymptotic for specific posets P [73, 82, 146, 95]. This chapter focuses on results that stem from our attempt to deepen our understanding of the parameter sat^*. Most of them directly concern sat^*, but some also could be of interest in other areas of mathematics. For example, we generalize a lemma from Lehman and Ron, which has multiple implications in the field of property testing. We also define new poset parameters raising new, interesting structural questions about posets, some of which are also addressed in this chapter.

Fixed poset Computing $\mathrm{sat}^*(n, P)$ appears to be a challenging task even for very simple posets P. For example, the asymptotic of $\mathrm{sat}^*(n, \diamondsuit)$ for $\diamondsuit$ the 4-element diamond poset (see Fig. 3.1) is still unknown, despite being the focus of multiple papers [136, 95]. The best-known bounds are given below.

Theorem 3.4 ([95]). *For any n, we have*

$$(2\sqrt{2} - o(1))\sqrt{n} \leqslant \mathrm{sat}^*(n, \diamondsuit) \leqslant n + 1.$$

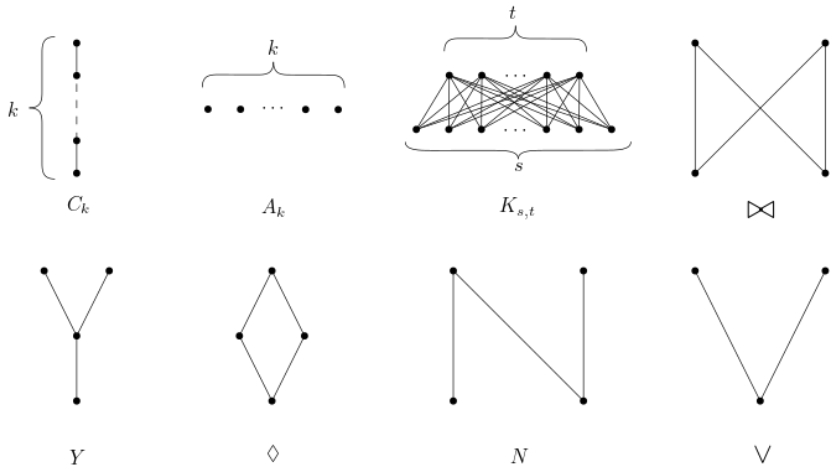


Figure 3.1: Hasse diagrams of some posets. Vertices represent elements of P. For $a, b \in P$, $a \leqslant b$ if and only if there exists a path from a to b only using upward edges.

Poset	$\mathrm{sat}^*(n, P)$	**References**
For all poset P	$\cdot = O(1)$ or $\cdot \geqslant 2\sqrt{n-2}$ $\cdot = O(n^{\lvert P\rvert})$	[77] [27]
$C_k, k \in [2,6]$	$\cdot = 2^{k-2}$	[82]
$C_k, k \geqslant 7$	$\sqrt{k}2^{k/2} \leqslant \cdot \leqslant 2^{\left(1-\frac{1}{5}\log_2 28\right)k}$	[139]
$A_k, k \in [2,6]$	$\cdot = n(k-1) - \Theta(k \log k)$	[28]
$\vee$ (fork), $\wedge$	$\cdot = n+1$	[73]
$K_{s,2}$	$n+1 \leqslant \cdot = O(s^3 n)$	[130]
$K_{s,t}$	$\cdot = O_{s,t}(n)$	[130]
$\Diamond$ (diamond)	$(2\sqrt{2} - o(1))n \leqslant \cdot \leqslant n+1$	[95]
$\bowtie$ (butterfly)	$n+1 \leqslant \cdot \leqslant \binom{n}{2} + 2n - 1$	[94] [73]
$Y, \vee + C_1$	$2\sqrt{n-2} \leqslant \cdot$	[116] [77]
N	$2\sqrt{n-2} \leqslant \cdot \leqslant 2n$	[94] [77, 138]
$2C_2, k \in [2,6]$	$\frac{3}{2}n + \frac{1}{2} \leqslant \cdot \leqslant 2n$	[116]
$3C_2, 5C_2, 7C_2$	$\cdot \leqslant 14, \cdot \leqslant 42, \cdot \leqslant 60$	[116]
$\Diamond^-$ (diamond minus an edge)	$= 4$	[116]
$C_3 + C_1$	$3 \leqslant . \leqslant 8$	[82]
UCTP with top chain (see [116] for def.)	$2\sqrt{n-2} \leqslant \cdot$	[116] [77]

Table 3.2: Summary of some bounds known for $\mathrm{sat}^*(n, P)$. This table is not exhaustive.

The diamond poset is not the only poset for which computing sat^* is challenging. Table 3.2 summarises the state of the art for some posets (the notations are explained in Fig. 3.1). For example, even in the simple case of C_k, the chain of size k (i.e. the total order on k elements), the exact value of $\mathrm{sat}^*(n, C_k)$ is still unknown, despite being the topic of multiple papers [82, 146, 139] (see Table 3.2 for the best-known bounds). Similarly, the antichain of size k, denoted A_k (i.e, the poset where every pair of elements is incomparable), has also been extensively studied. Ferrara, Kay, Kramer, Martin, Reiniger, Smith and Sullivan [73] were the first to study $\mathrm{sat}^*(n, A_k)$ and made the following conjecture.

Conjecture 3.5 ([73]). *For $k \geqslant 3$, $\mathrm{sat}^*(n, k) \sim (k-1)n$ as $n \to \infty$.*

Note that the upper bound is easy to see: for all $i \in [n]$, a k-antichain saturated family can contain at most $k-1$ subsets of size i since two subsets of the same size are incomparable. A few years later, Martin, Smith, and Walker [136] proved the following bound.

Theorem 3.6 ([136]). *For any $k \geqslant 4$ and n sufficiently large compared to k,*

$$\mathrm{sat}^*(n, A_k) \geqslant \left(1 - \frac{1}{\log_2(k-1)}\right) \frac{(k-1)n}{\log_2(k-1)}.$$

The exact values for $k = 2, 3$ and 4 were shown to be $n+1$, $2n$ and $3n-1$ respectively in [73], the exact values for $k = 5$ and $k = 6$ were recently determined to be $4n-2$ and $5n-5$ respectively by Đanković and Ivan [15]. They also strengthened Conjecture 3.5 as follows, and proposed two weaker conjectures implied by Conjecture 3.7.

Conjecture 3.7 ([15]). $\mathrm{sat}^*(n, k) = n(k-1) - O_k(1)$.

We answer positively to both conjectures and also compute a closed formula for the exact value of $\mathrm{sat}^*(n, A_k)$. We would like to note that this is the first infinite family of posets for which sat^* is known exactly. To state our result we need to define the following notions[1]. Given a fixed n, $\mathcal{C}(m, t)$ denotes the initial segment of size m in $\binom{[n]}{t}$ when the sets are ordered in colexicographically. For a family of sets $\mathcal{A} \subseteq \binom{[n]}{t}$, let $\nu(\mathcal{A})$ be the size of the maximum matching from $\mathcal{A}$ to its shadow $\partial\mathcal{A}$, and recursively define $c_0, c_1, \ldots, c_{\lfloor \ell/2 \rfloor}$ as follows. Let $c_{\lfloor \ell/2 \rfloor} = k-1$. For $0 \leqslant t < \lfloor \ell/2 \rfloor$, let $c_t = \nu\left(\mathcal{C}(c_{t+1}, t+1)\right)$.

Theorem 3.8. *Let $n, k \geqslant 4$ be integers and let ℓ and $c_0, \ldots, c_{\lfloor \ell/2 \rfloor}$ be as defined above. If $n < \ell$, then $\mathrm{sat}^*(n, k) = 2^n$. If $n \geqslant \ell$, then*

$$\mathrm{sat}^*(n, k) \geqslant 2 \sum_{t=0}^{\lfloor \ell/2 \rfloor} c_t + (k-1)(n - 1 - 2\lfloor \ell/2 \rfloor).$$

Moreover, equality holds when $n \geqslant 2\ell + 1$.

[1] The notions of shadow and colexicographical order are formally defined in Section 3.2.

A generalisation of Lehman and Ron's lemma While working toward Theorem 3.8, we became interested in the structure of chain decompositions in the Boolean lattice. We proved a generalisation of a lemma by Lehman and Ron, which has been tremendously useful in the field of property testing [128]. We believe this generalisation to be of interest in its own right.

To state Lehman and Ron's lemma we need to introduce the notion of *skipless* chain. We say that a chain $C_1 \subsetneq C_2 \subsetneq \ldots \subsetneq C_r \subseteq [n]$ is *skipless* if it has the property $|C_{i+1}| = |C_i| + 1$ for all $i \in [r-1]$. This concept has also been studied in other contexts such as in [18, 66, 132].

Theorem 3.9 (Lehman-Ron [128]). *Let integers $1 \leqslant s < r \leqslant n$ be given and for all $i \in [m]$, subsets $X_i \subseteq Y_i \subseteq [n]$ with $|X_i| = s$ and $|Y_i| = r$. Then there exist m disjoint skipless chains that cover $\{X_1, \ldots, X_m, Y_1, \ldots, Y_m\}$.*

Our generalisation comes from a natural question. What happens if we allow the sets to come from different layers, or ask that the chains go via some elements from layers between layer r and layer s? Is it possible to cover any m disjoint chains with m disjoint skipless chains, or can we force the use of an additional chain? We show that m chains always suffice.

Theorem 3.10. *Suppose that $\mathcal{F} \subseteq 2^{[n]}$ admits a chain decomposition into m chains. Then there exist disjoint skipless chains $C^1, \ldots, C^m \subseteq 2^{[n]}$ such that $\mathcal{F} \subseteq \bigcup_{i=1}^{m} C^i$.*

To showcase the strength of this generalisation, we provide a proof of the following corollary, which is weaker than Theorem 3.8 but still resolves both Conjecture 3.5 and Conjecture 3.7.

Corollary 3.11. *There exist constants $c_1, c_2 > 0$ such that for all $k \geqslant 4$ and n sufficiently large,*

$$n(k-1) - c_1 k \log k \leqslant \mathrm{sat}^*(n,k) \leqslant n(k-1) - c_2 k \log k.$$

The upper bound part was already known [15]. So the following proof only includes an argument for the lower bound.

Proof. Recall that ℓ is the smallest j such that $\binom{j}{\lfloor j/2 \rfloor} \geqslant k-1$, so $\ell = \Theta(\log k)$. By Dilworth's theorem [61] (see Theorem 0.2), having a chain decomposition of size at most $k-1$ is equivalent to not containing any antichain of size k. Suppose that $\mathcal{F} \subseteq 2^{[n]}$ is k-antichain saturated and so admits a decomposition into $k-1$ chains. By Theorem 3.10, there are $k-1$ disjoint skipless chains $C^1, \ldots, C^{k-1}$ that cover the elements of $\mathcal{F}$; since $\mathcal{F}$ is saturated, this must form a chain decomposition of $\mathcal{F}$. It suffices to show that every chain must contain a set of size at most ℓ and a set of size at least $n - \ell$. Suppose the smallest element X of some chain C^i has size $|X| > \ell$, then all subsets Y of X must be present in $\mathcal{F}$ since otherwise

we may extend C^i to include Y (and that would mean that $\mathcal{F} \cup \{Y\}$ can also be covered by $k-1$ chains, contradicting the fact that $\mathcal{F}$ is k-antichain saturated). There are at least $k-1$ subsets of X of size $\lfloor \ell/2 \rfloor$, and these cannot all be covered by the other $k-2$ chains. Since each chain contains an element of size at most ℓ and one of size at least $n-\ell$, the bound follows immediately from the fact that the chains are skipless. □

In order to prove the exact lower bound of Theorem 3.8, we would need to examine what happens on layers $1, \dots, \ell$. This is considerably more delicate, and the full proof is not included in this manuscript due to space constraints. In particular, we also needed to design an explicit construction of a k-antichain saturated system $\mathcal{F}$ which matches our lower bound, provided n is sufficiently large. This construction was already known for the special case $k-1 = \binom{\ell}{\lfloor \ell/2 \rfloor}$, and we apply it recursively for other values of k. The recursion requires special care and depends on a particular way of writing $k-1$ as a sum of binomial coefficients. This notation can be used to write exact values for the matching numbers c_t from Theorem 3.8. In Section 3.4.2, we describe our explicit construction but omit the proof that this family is indeed A_k-saturated.

General theorems Although sat^* has been investigated for various specific posets, very few general results are known to hold, and as seen above, many fundamental questions remain open. Nonetheless, the study of general properties of sat^* has gained significant momentum in recent years, yielding promising results [77, 27, 96]. It was first shown that the growth of $\mathrm{sat}^*(n, P)$ has a dichotomy. Keszegh, Lemons, Martin, Pálvölgyi and Patkós [116] proved that for any poset P, $\mathrm{sat}^*(n, P)$ is either bounded or at least $\log_2(n)$, and they made the following conjecture.

Conjecture 3.12 ([116]). *For any fixed poset P, and any $n \in \mathbb{N}$, either*

$$\mathrm{sat}^*(n, P) = O(1) \qquad \textit{or} \qquad \mathrm{sat}^*(n, P) \geqslant n + 1.$$

Note that the conjecture above, even in the very restricted case where P is the 4-element diamond poset, is not yet known to hold. Nonetheless, a few years later, progress was made towards this conjecture. Freschi, Piga, Sharifzadeh and Treglown [77] proved the following result.

Theorem 3.13 ([77]). *For any fixed poset P and any $n \in \mathbb{N}$, either*

$$\mathrm{sat}^*(n, P) = O(1) \qquad \textit{or} \qquad \mathrm{sat}^*(n, P) \geqslant 2\sqrt{n-2}.$$

Surprisingly, there is no known poset P for which $\mathrm{sat}^*(n, P) = \omega(n)$. In fact, some researchers believe in an even stronger version of Conjecture 3.12 written below.

Conjecture 3.14. *For any fixed poset P and any $n \in \mathbb{N}$, either*

$$\mathrm{sat}^*(n, P) = O(1) \qquad \textit{or} \qquad \mathrm{sat}^*(n, P) = \Theta_P(n).$$

Whilst as summarised above, some general lower bounds have been established, prior to our work, no non-trivial general upper bounds have yet been found. We showed that the saturation number has at worst polynomial growth.

Theorem 3.15. *For any fixed poset P and any $n \in \mathbb{N}$, $\mathrm{sat}^*(n, P) = O(n^c)$, where $c \leqslant |P|$ is a constant depending on P only.*

To prove Theorem 3.15 we introduce two new key notions, "cube-height" and "cube-width". Intuitively, for a poset P, the "cube-height" is the least k such that, for some n, we can embed P into the first $k+1$ layers of Q_n, while the "cube-width" is the smallest n that makes such a "small height" embedding possible. See Fig. 3.3 for a few examples.

Definition 3.16. *For a poset P, the cube-height $\mathrm{h}^*(P)$ is the minimum $h^* \in \mathbb{N}$ for which there exists $n \in \mathbb{N}$ such that $\binom{[n]}{\leqslant h^*}$ contains an induced copy of P.*

Definition 3.17. *For a poset P, the cube-width $\mathrm{w}^*(P)$ is the minimum $w^* \in \mathbb{N}$ such that there exists an induced copy of P in $\binom{[w^*]}{\leqslant \mathrm{h}^*(P)}$.*

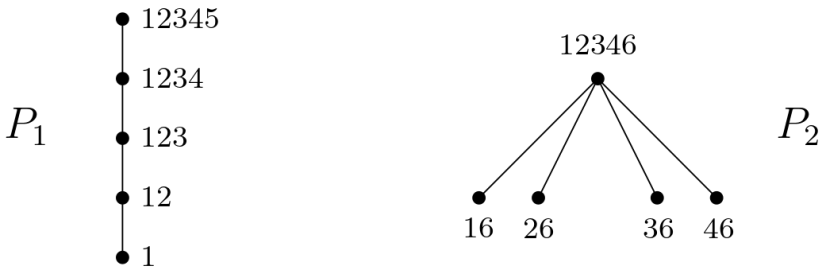


Figure 3.3: Hasse diagram of two posets P_1 and P_2 such that $\mathrm{h}^*(P_1) = 4$, $\mathrm{w}^*(P_1) = 4$ and $\mathrm{h}^*(P_2) = 2, \mathrm{w}^*(P_2) = 4$. We also represented a witness assignement[3] for $P_1 + P_2$ achieving optimal cubeheight and cubewidth, $\mathrm{h}^*(P_1 + P_2) = 5$ and $\mathrm{w}^*(P_1 + P_2) = 6$.

Theorem 3.15 is in fact a direct consequence of the two following theorems.

Theorem 3.18. *For any fixed poset P and any $n \geqslant 2|P|$, $\mathrm{sat}^*(n, P) = O(n^{\mathrm{w}^*(P)})$.*

Theorem 3.19. *For any poset P, $\mathrm{w}^*(P) \leqslant |P|$.*

[3] abc stands for the set $\{a, b, c\}$ – we use this convention in throughout this section.

Our construction in Theorem 3.18 could be interpreted as the result of a greedy algorithm where the sets are ordered according to size (and then arbitrarily within the layers). An element is added to the family as long as it does not create a copy of P inside the current family. Greedy algorithms have been used before for studying poset saturation; most notably, a greedy colex algorithm was used to show a linear upper bound for the butterfly [116]. Our result shows that "layer-by-layer" greedy algorithms result in a saturated family of size at most $n^{|P|}$, and we note that such an algorithm has a near-linear time complexity of $O_P(|Q_n|(\log_2|Q_n|)^{|P|^2}) = O_P(2^n n^{|P|^2})$. This follows from the fact that for any family $\mathcal{F}$, it can be decided if it is P-free in $O_P(|\mathcal{F}|^{|P|})$ time.

Roadmap In Section 3.2, we define the important notions and tools used in this chapter. Section 3.3 aims to provide a complete proof of Theorem 3.15. This section is split into two subsections: first, in Section 3.3.1, we prove Theorem 3.18, and then, in Section 3.3.2, we prove Theorem 3.19. Section 3.4.1 focuses on Theorem 3.10. Finally, in Section 3.4.2, we give an overview of our exact bound for the antichain saturation number stated in Theorem 3.8.

3.2 Preliminaries

Colex for shadows and matchings In the *colexicographic* or *colex* order on $\binom{[n]}{t}$, we have $A < B$ if $\max(A \triangle B) \in B$, where $\triangle$ denotes the symmetric difference $A \triangle B = (A \setminus B) \cup (B \setminus A)$. Informally, sets with larger elements come later in the order. For $t = 3$ the initial segment of size 8 in colex is given by

$$\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,5\}, \{1,3,5\}, \{2,3,5\}, \{1,4,5\}.$$

We write $\mathcal{C}(m,t)$ for the initial segment of colex on layer t of size m.

For a family of sets $A \subseteq \binom{[n]}{t}$, the *shadow* of $\mathcal{A}$ is given by

$$\partial\mathcal{A} = \{X \in \tbinom{[n]}{t-1} \mid X \subseteq Y \text{ for some } Y \in \mathcal{A}\}.$$

The well-known Kruskal-Katona theorem below shows that the shadow of a family of subsets of size t is minimised by taking the family to be an initial segment of colex, and we will prove an analogous result about matchings between a family and its shadow.

Theorem 3.20 (Kruskal-Katona [127]). *Let $1 \leqslant t \leqslant n$ be integers. Let $\mathcal{B} \subseteq \binom{[n]}{t}$ and let $\mathcal{C}$ be the initial segment of colex on $\binom{[n]}{t}$ of size $|\mathcal{B}|$. Then $|\partial\mathcal{B}| \geqslant |\partial\mathcal{C}|$.*

For $\mathcal{B} \subseteq \binom{[n]}{t}$, let $\nu(\mathcal{B})$ denote the size of the maximum matching in the bipartite graph between $\mathcal{B}$ and $\partial\mathcal{B}$, where $X \in \mathcal{B}$ is adjacent to $Y \in \partial\mathcal{B}$ if $Y \subseteq X$.

Lemma 3.21. *Let $1 \leqslant t \leqslant n$ be integers. Let $\mathcal{B} \subseteq \binom{[n]}{t}$ and let $\mathcal{C}$ be the initial segment of colex on $\binom{[n]}{t}$ of size $|\mathcal{B}|$. Then $\nu(\mathcal{B}) \geqslant \nu(\mathcal{C})$.*

We omit the proof in this manuscript, but it is available in [28] (Lemma 2.4).

Cascade notation Let m, r be integers. For our upper bound construction, we need a result that gives the value of $\nu(\mathcal{C}(m,r))$.

There is a unique way of writing m as

$$m = \binom{a_r}{r} + \binom{a_{r-1}}{r-1} + \cdots + \binom{a_s}{s}$$

where $r \geqslant s \geqslant 1$, $a_r > \cdots > a_s > 0$ and $a_i \geqslant i$ for all $i \in [s]$. The initial segment of colex $\mathcal{C}(m,r)$ of size m on layer r is the union of the set $\binom{[a_r]}{r}$, the set containing all elements of the form $A \cup \{a_r + 1\}$ with $A \in \binom{[a_{r-1}]}{r-1}$, the set containing all elements of the form $A \cup \{a_r+1, a_{r-1}+1\}$ where $A \in \binom{[a_{r-2}]}{r-2}$, and so on until the sets containing all the elements of the form $A \cup \{a_r + 1, a_{r-1} + 1, \ldots, a_{s+1} + 1\}$ where $A \in \binom{[a_s]}{s}$.

The expansion above is also called the *r-cascade notation* of m and may be built recursively as follows. Take a_r to be the largest j such that $\binom{j}{r} \leqslant m$, and set $m' = m - \binom{j}{r}$. If $m' = 0$, the recursion ends. Otherwise, we append the expansion for m' and $r' = r - 1$.

This expansion can be used to compute the size of the shadow $|\partial\mathcal{C}(m,r)|$, but we are interested in using it to give the precise value of $\nu(\mathcal{C}(m,r))$ as follows.

Lemma 3.22. *Let $r \geqslant s \geqslant 1$ and $a_r > \cdots > a_s > 0$ be such that*

$$m = \binom{a_r}{r} + \binom{a_{r-1}}{r-1} + \cdots + \binom{a_s}{s}. \tag{3.1}$$

If $i \leqslant \lceil a_i/2 \rceil$ for all $i \in [s,r]$, then $\nu(\mathcal{C}(m,r)) = \sum_{i=s}^{r} \binom{a_i}{i-1}$. Otherwise, let $j \in [s,r]$ be maximal such that $j > \lceil a_j/2 \rceil$. Then

$$\nu(\mathcal{C}(m,r)) = \binom{a_r}{r-1} + \cdots + \binom{a_{j+1}}{j} + \binom{a_j}{j} + \cdots + \binom{a_s}{s}.$$

The proof of Lemma 3.22 is omitted in this manuscript but available in [28].

VC-dimension We say that a family $\mathcal{F}$ of subsets of $[n]$ *shatters* a set $S \subseteq [n]$ if, for all $F \subseteq S$, there exists $A \in \mathcal{F}$ such that $A \cap S = F$. In other words, $\{A \cap S : A \in \mathcal{F}\}$ is the power set of S. The *VC-dimension* of $\mathcal{F}$ is the largest cardinality of a set shattered by $\mathcal{F}$. The size of a family $\mathcal{F}$ with bounded VC-dimension grows at worst polynomially, as shown by the following well-known result.

Lemma 3.23. *[Sauer-Shelah lemma [166, 168]] If $\mathcal{F} \subseteq 2^{[n]}$ has VC-dimension d then,*

$$|\mathcal{F}| \leqslant \sum_{i=0}^{d} \binom{n}{i}.$$

3.3 A universal upper bound through cubewidth

The aim of this section is to prove Theorem 3.15. It is divided into two subsections. First, we focus on the saturation part of the theorem and prove Theorem 3.18. Then, we investigate the parameter *cube-width* and prove Theorem 3.19.

3.3.1 A universal upper bound

This section focuses on Theorem 3.18. The proof is fairly short and relies on the use of the VC-dimension introduced in the preliminaries, along with two new poset parameters introduced earlier (see Definition 3.16 and Definition 3.17).

Theorem 3.18. *For any fixed poset P and any $n \geqslant 2|P|$, $\mathrm{sat}^*(n, P) = O(n^{\mathrm{w}^*(P)})$.*

Proof of Theorem 3.18. Let $h^* = \mathrm{h}^*(P)$, $w^* = \mathrm{w}^*(P)$, and assume $n \geqslant 2w^*$. Let $\mathcal{F}_0$ be the family consisting of the first h^* layers, or in other words, all the elements of size at most $h^* - 1$. By the definition of the cube-height, the family $\mathcal{F}_0$ does not contain an induced copy of P. We now extend this family to a P-saturated family in an arbitrary fashion. Let $\mathcal{F}$ be this resulting family. The crucial property of this family is the following.

Claim 3.24. *The VC-dimension of $\mathcal{F}$ is less than w^*.*

Proof. Suppose towards a contradiction that $\mathcal{F}$ shatters a set S of size w^*. By definition, this means that $\mathcal{L} = \{A \cap S : A \in \mathcal{F}\}$ is the power set of S, and it is isomorphic to Q_{w^*}. Since w^* is the cube-width of P, we can find a copy of P in $\mathcal{L}$ such that all sets have size at most h^*. Let P_S denote such a copy of P in $\mathcal{L}$.

Let $M_1, \ldots, M_s$ be the sets in P_S that have size exactly h^* – they are subsets of S by construction. Let $P' = P_S \setminus \{M_1, \ldots, M_s\}$. Since we have removed all elements of P_S of maximal size, the cube-height of P' is strictly less than that of P_S (i.e. $\mathrm{h}^*(P') < \mathrm{h}^*(P_S) = \mathrm{h}^*(P)$). Therefore there exist an embedding of P' in the first h^* layers. These layers are part of $\mathcal{F}_0$, thus P' is contained in $\mathcal{F}_0 \subseteq \mathcal{F}$.

Since each M_i is a subset of S, we can find $A_i \in \mathcal{F}$ such that $A_i \cap S = M_i$ for all $i \leqslant s$. Note that this implies that $|A_i| \geqslant h^*$ for all $i \leqslant s$, and so no A_i appears in P'. We now show that $P' \cup \{A_1, \ldots, A_s\}$ is an induced copy of P in $\mathcal{F}$, which will yield the desired contradiction.

First, if $B \in P'$ is incomparable to M_i, then B is also incomparable to A_i. This is because if B is a subset of A_i, then it is also a subset of $A_i \cap S = M_i$, a contradiction. Conversely, if $B \in P'$ is a subset of $M_i = A_i \cap S$, then it is a subset of A_i, too. We also have that A_i and A_j are incomparable for $i \neq j$ as they are incomparable when restricted to S. Finally, A_i can never be a subset of $B \in P'$, since $A_i \cap S = M_i$ is not a subset of $B \cap S = B$.

We conclude that $P' \cup \{A_1, \ldots, A_s\}$ is an induced copy of P in $\mathcal{F}$. This contradict our intial assumption that $\mathcal{F}$ is P-satured and therefore P-free, proving that the VC-dimension of $\mathcal{F}$ is strictly less than w^*, as desired. □

Combining Lemma 3.23 and Claim 3.24, we conclude that, as $n \geqslant 2w^*$,

$$\mathrm{sat}^*(n, P) \leqslant |\mathcal{F}| \leqslant \sum_{i=0}^{w^*-1} \binom{n}{i} \leqslant w^* \frac{n^{w^*-1}}{(w^*-1)!} \leqslant 2n^{w^*-1}.$$

Here we have used that $\frac{m}{(m-1)!} \leqslant 2$ for all $m \in \mathbb{N}$, and that, since $n \geqslant 2w^*$, the largest binomial coefficient in the above sum is $\binom{n}{w^*-1}$. □

3.3.2 Cubewidth

This section explores the new concept of *cube-width* and related notions in structural poset theory. One of the consequences of our study is a tight bound on the value that *cube-width* can take. In particular, Theorem 3.19 is derived directly from the main theorem of this section.

Inclusion representation Before stating this theorem, we define *inclusion representation* and introduce the concepts of equivalent and irreducible *inclusion representations*.

In the following, a poset is called *trivial* if it has exactly one element and a poset is *non-trivial* if it has more than one element. Given a poset P, a family $\mathcal{S} = \{S_x : x \in P\}$ of sets is called an *inclusion representation* of P if for all $x, y \in P$, we have $x \leqslant y$ in P if and only if $S_x \subseteq S_y$. We say that an inclusion representation $\mathcal{S}$ *uses* $|\bigcup \mathcal{S}|$ elements.

With these definitions the *cube-height* of a poset P can be seen as the least non-negative integer h such that P has an inclusion representation $\mathcal{S} = \{S_x : x \in P\}$ where $|S_x| \leqslant h$ for every $x \in P$. Similarly the *cube-width* of a poset P is the least non-negative integer w for which there is an inclusion representation $\mathcal{S}$ of P using at most w elements such that $|S_x| \leqslant \mathrm{h}^*(P)$ for every $x \in P$.

Note that every poset has an inclusion representation. Indeed, for a poset P and an element $x \in P$, we denote by $D_P[x]$, the *closed downset* of x in P, that is, the set of all $y \in P$ such that $y \leqslant x$ in P. Now, it is easy to verify that $\{D_P[x] : x \in P\}$ is an inclusion

representation of P. We call this representation the *canonical inclusion representation*. Let P be a poset, and let $\mathcal{S} = \{S_x : x \in P\}$ and $\mathcal{S}' = \{S'_x : x \in P\}$ be inclusion representations of P. We say that $\mathcal{S}$ and $\mathcal{S}'$ are *isomorphic* if $|\bigcup \mathcal{S}| = |\bigcup \mathcal{S}'|$ and there is a bijection $f : \bigcup \mathcal{S} \to \bigcup \mathcal{S}'$ such that for every $x \in P$ and every $a \in \bigcup \mathcal{S}$, $a \in S_x$ if and only if $f(a) \in S'_x$. We say that $\mathcal{S}'$ is a *reduction* of $\mathcal{S}$ if $|\bigcup \mathcal{S}'| \leqslant |\bigcup \mathcal{S}|$ and $|S'_x| \leqslant |S_x|$ for every $x \in P$. With this definition, an inclusion representation is a reduction of itself. We say that $\mathcal{S}$ is *equivalent* to $\mathcal{S}'$ if $\mathcal{S}$ is a reduction of $\mathcal{S}'$ and $\mathcal{S}'$ is a reduction of $\mathcal{S}$. Note that two inclusion representations of a poset can be equivalent but not isomorphic – see Figure 3.4. We say that $\mathcal{S}'$ is a *strict reduction* of $\mathcal{S}$ if $\mathcal{S}'$ is a reduction of $\mathcal{S}$ and they are not equivalent. An inclusion representation that has no strict reduction is said to be *irreducible*. In Figure 3.5, we give an example of a poset with two irreducible inclusion representations $\mathcal{S}$ and $\mathcal{S}'$ with $|\bigcup \mathcal{S}| \neq |\bigcup \mathcal{S}'|$.

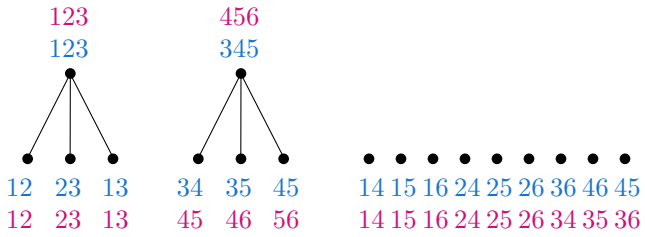


Figure 3.4: The Hass diagram of a 17-elements poset P, with two inclusion representations of P (in blue and in pink). These representations are equivalent but not isomorphic.

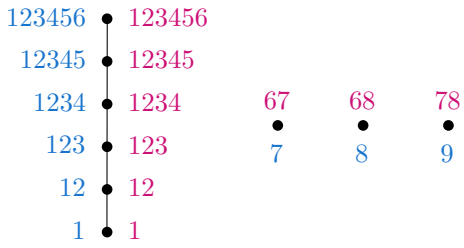


Figure 3.5: Th Hasse diagram of a 9-elements poset P. And two irreducible configuration. $\mathrm{w}^*(P) = 8$ and $\mathrm{iir}(P) = 9$.

A natural question that arises is how many elements irreducible inclusion representations can use. Thus, for a poset P, we define $\mathrm{iir}(P)$ as the maximum non-negative integer w such that there is an irreducible inclusion representation of P that uses w elements. Note that, for every poset P, we have $\mathrm{w}^*(P) \leqslant \mathrm{iir}(P)$. Indeed, for a poset P, take an inclusion representation $\mathcal{S}$ of P witnessing $\mathrm{h}^*(P)$ and then set $\mathcal{S}'$ to be an irreducible inclusion

representation of P such that $\mathcal{S}'$ is a reduction of $\mathcal{S}$. Therefore, the following result implies Theorem 3.19.

Theorem 3.25. *Every irreducible inclusion representation of every poset P uses at most $|P|$ elements. In other words,* $\mathrm{iir}(P) \leqslant |P|$ *for every poset P.*

The theorem admit a fairly easy proof through induction, once the induction hypothesis is stated cleverly. the following lemma encapsulate the key property for this induction.

Lemma 3.26. *Let P be a poset, let $\mathcal{S} = \{S_x : x \in P\}$ be an inclusion representation of P, and let $y \in P$. Let $Q = P - D_P[y]$, and let Q' be a poset with the ground set $\{S_x - S_y : x \in P - \{y\}\}$ equipped with the inclusion relation. Let $\varepsilon \in \{0, 1\}$ be the number of unique minimal elements of Q'. Then, there exists an inclusion representation $\mathcal{S}' = \{S'_x : x \in P\}$ of P with $|S'_x| \leqslant |S_x|$ for every $x \in P$ and*

$$\left|\bigcup \mathcal{S}'\right| \leqslant \mathrm{iir}(Q') + \varepsilon + |S_y|.$$

Proof. Let $A = \emptyset$ when Q' has no unique minimal element and $A = \zeta$ when ζ is a unique minimal element of Q' (note that in this case $\zeta \neq \emptyset$ since otherwise $q \leqslant y$ in P where $q \in P$ is such that $S_q - S_x = \zeta$). Clearly, $\mathcal{T}' = \{\alpha - A : \alpha \in Q'\}$ is an inclusion representation of Q'. Let $\mathcal{R}' = \{R'_\alpha : \alpha \in Q'\}$ be an irreducible inclusion representation of Q' that is a reduction of $\mathcal{T}'$. In particular, $\mathcal{R}'$ uses at most $\mathrm{iir}(Q')$ elements. Assume that the ground sets of $\mathcal{S}$ and $\mathcal{R}'$ are disjoint. For every $x \in D_P[y]$, let $S'_x = S_x$. Choose $A' \subset A$ arbitrarily so that $|A'| \leqslant 1$ (note that $\varepsilon = |A'|$). For every $x \in P - D_P[y]$, let $S'_x = R'_\alpha \cup (S_x \cap S_y) \cup A'$ where $\alpha \in Q'$ is such that $\alpha = S_x - S_y$. We claim that $\mathcal{S}' = \{S'_x : x \in P\}$ is an inclusion representation of P and $|S'_x| \leqslant |S_x|$ for every $x \in P$.

Let $x, z \in P$. If $x \leqslant z$ in P, then $S'_x \subset S'_z$ by definition. Thus, assume $x \not\leqslant z$ in P. If $x, z \in D_P[y]$, then $S'_x \not\subset S'_z$ as $\mathcal{S}$ is an inclusion representation of P. If $x, z \notin D_P[y]$, then either $(S_x \cap S_y) \not\subset (S_z \cap S_y)$ or $(S_x - S_y) \not\subset (S_z - S_y)$. In the former case clearly $S'_x \not\subset S'_z$ and in the latter case we have $R'_\alpha \not\subset R'_\beta$, where $\alpha = S_x - S_y$ and $\beta = S_z - S_y$, and so, $S'_x \not\subset S'_z$. Next, suppose that $x \in D_P[y]$ and $z \notin D_P[y]$. If $S'_x \subset S'_z$, then $S_x \subset S_z \cap S_y \subset S_z$, which is not possible. Finally, assume $x \notin D_P[y]$ and $z \in D_P[y]$. If Q' has no unique minimal element, then $R_\alpha \neq \emptyset$ for every $\alpha \in Q'$, and so, $S'_x \not\subset S'_z$. Otherwise, $A' \subset S'_x - S'_z$, which implies $S'_x \not\subset S'_z$. The above case analysis yields that $\mathcal{S}'$ is indeed an inclusion representation of P. Now, we argue that $\mathcal{S}'$ is a reduction of $\mathcal{S}$. For every $x \in D_P[y]$, we have $|S'_x| = |S_x|$, and for every $x \in P - D_P[y]$, we have

$$|S'_x| = |R'_\alpha| + |S_x \cap S_y| + |A'| \leqslant |\alpha - A| + |S_x \cap S_y| + |A'| = |S_x - S_y| - |A| + |S_x \cap S_y| + |A'| \leqslant |S_x|.$$

Finally,

$$\left|\bigcup \mathcal{S}'\right| \leqslant \left|\bigcup \mathcal{R}'\right| + |A'| + |S_y| \leqslant \mathrm{iir}(Q') + \varepsilon + |S_y|. \qquad \square$$

Proof of Theorem 3.25. The proof is by induction on the number of elements of P. If P is trivial, then the statement is clear. Suppose that P is non-trivial and let $\mathcal{S} = \{S_x : x \in P\}$ be an irreducible inclusion representation of P. Additionally, suppose to the contrary that $\mathcal{S}$ uses more than $|P|$ elements. If $|D_P[x]| \leqslant |S_x|$ for every $x \in P$, then the canonical inclusion representation of P is a strict reduction of $\mathcal{S}$, which contradicts the irreducibility of $\mathcal{S}$.

Therefore, we can assume that there is $y \in P$ with $|D_P[y]| > |S_y|$. Observe that y is not a unique maximal element in P as otherwise $|S_y| = |\bigcup \mathcal{S}| > |P| = |D_P[y]|$. Hence, y is not a unique maximal element, and so, $Q = P - D_P[y]$ is non-empty. Consider a poset Q' with the ground set $\{S_x - S_y : x \in Q\}$ equipped with the inclusion relation and let $\varepsilon \in \{0, 1\}$ be the number of unique minimal elements of Q'. Note that by induction $\operatorname{iir}(Q') \leqslant |Q'|$. By Lemma 3.26, there is an inclusion representation $\mathcal{S}' = \{S'_x : x \in P\}$ of P with $|S'_x| \leqslant |S_x|$ for every $x \in P$ and

$$\left|\bigcup \mathcal{S}'\right| \leqslant \operatorname{iir}(Q') + \varepsilon + |S_y| \leqslant |Q'| + 1 + (|D_P[y] - 1) \leqslant |P| < \left|\bigcup \mathcal{S}\right|.$$

This shows that $\mathcal{S}'$ is a strict reduction of $\mathcal{S}$, which is a contradiction. □

3.4 Antichain saturation through Lehman and Ron's lemma

The aim of this section is to discuss our results on antichain saturation Theorem 3.8. To understand the structure of an A_k-saturated family, where A_k denotes the antichain of size k, we first need to examine the structure of chains in the Boolean lattice. Our study led to Theorem 3.10, a generalisation of Lehman and Ron's lemma [128]. We believe this generalisation could be of interest beyond the scope of saturation theory.

3.4.1 A generalisation of Lehman and Ron's lemma

For convenience, we recall the statement of Theorem 3.10 below.

Theorem 3.10. *Suppose that $\mathcal{F} \subseteq 2^{[n]}$ admits a chain decomposition into m chains. Then there exist disjoint skipless chains $C^1, \dots, C^m \subseteq 2^{[n]}$ such that $\mathcal{F} \subseteq \bigcup_{i=1}^m C^i$.*

The following lemma uses the main building block in the inductive argument we will employ to prove Theorem 3.10.

Lemma 3.27. *Let $s \leqslant r \leqslant n$ be integers. Let $C^1, \dots, C^m$ be disjoint chains, such that for all $i \in [m-1]$, the chain C^i starts in layer s and ends in layer r. Suppose that C^m starts in $A \in \binom{[n]}{\leqslant s}$ and ends in $B \in \binom{[n]}{r}$. Then there exist m disjoint chains $D^1, \dots, D^m$ with the following three properties.*

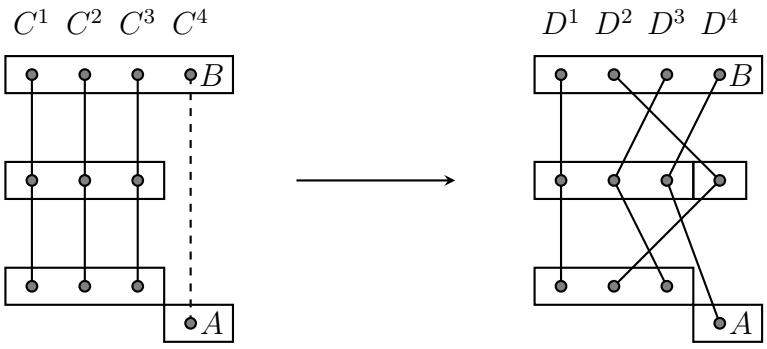


Figure 3.6: Representation of Lemma 3.27 (case $r = s + 2$ and $m = 4$).

1. *For $i \in [m-1]$, the chain D^i starts in the sth layer, ends in the rth layer and is skipless.*
2. *The chain D^m starts at A and intersects the ith layer for all $i \in [s+1, r]$.*
3. *The chains $D^1, \dots, D^m$ cover the elements in $C^1, \dots, C^m$.*

Note that our lemma allows the element A to appear on a lower layer than the others (illustrated in Figure 3.6) and that it may be impossible to add an element on layer s to the chain D^m.

The overall structure of the proof of Lemma 3.27 is very similar to that of Lehman-Ron [128]. We first consider the special case in which $s = r - 2$. As in the proof of Lehman-Ron [128], the first step is to show that there are at least m elements in the $(r-1)$th layer that could be elements of the chains $D^1, \dots, D^m$.

Lemma 3.28. *Let a, r, n be integers satisfying $a \leqslant r - 2 \leqslant n - 2$, let $\mathcal{R} \subseteq \binom{[n]}{r}$ be of size m, let $\mathcal{S} \subseteq \binom{[n]}{r-2}$ be of size $m-1$ and let $A \in \binom{[n]}{a}$. Suppose that there exists a bijection $f : \mathcal{R} \to \mathcal{S} \cup \{A\}$ with $f(X) \subseteq X$ for all $X \in \mathcal{R}$.*
Let $\mathcal{Q}$ denote the set of $Q \in \binom{[n]}{r-1}$ with $S \subseteq Q \subseteq R$ for some $(S, R) \in (\mathcal{S} \cup \{A\}) \times \mathcal{R}$. Then $|\mathcal{Q}| \geqslant m$.

Proof. We prove the claim by contradiction. Consider a counterexample to the claim for which m is minimal. If $m = 1$, then we are given elements $A \subseteq R$ with $|A| \leqslant r - 2$ and $|R| = r$. Then there exists at least one element $Q \in \mathcal{Q}$ such that $A \subseteq Q \subseteq R$: simply remove one of the elements in $R \setminus A$ from R to obtain Q. We therefore assume $m \geqslant 2$.

We consider the Hasse diagram $H = (V, E)$ of $2^{[n]}$. Note that $\mathcal{Q}$ can be seen as the set of all elements of cardinality $r - 1$ lying on a path between an element of $\mathcal{S} \cup \{A\}$ and an element of $\mathcal{R}$.

We consider the "restriction" $H' = (V', E')$ which is obtained by taking the subgraph of H on vertex set $V' = \mathcal{R} \cup \mathcal{S} \cup \mathcal{Q} \cup \{A\}$, removing all arcs containing A and then adding an arc from A to Q for all $Q \in \mathcal{Q}$ with $A \subseteq Q$. We denote by $N^+(X)$ (resp. $N^-(X)$) the set of vertices Y with an arc $X \to Y$ (resp. with an arc $Y \to X$) in H', and define $d^+(X) = |N^+(Y)|$ and $d^-(X) = |N^-(X)|$. We first prove the following three claims.

Claim 3.29. *For every $R \in \mathcal{R}$ and every $Q \in N^-(R)$, we have $d^-(R) \geqslant d^-(Q)$.*

Proof. In order to prove the claim, for any R in $\mathcal{R}$, and any Q in $N^-(R)$, we exhibit an injective function $\pi : N^-(Q) \to N^-(R)$.

We denote by j the unique element of the set $R \setminus Q$. For $S \in \mathcal{S} \cap N^-(Q)$, we denote by i the unique element in $Q \setminus S$ and set $\pi(S) = R \setminus \{i\} = S \cup \{j\}$. Note that $\pi(S) \in \mathcal{Q}$ as $S, \pi(S), R$ is a path in H'. By doing so, we specified a unique $\pi(S) \in N^-(R)$ for all $S \in N^-(Q)$ except for possibly A if $A \in N^-(Q)$. However, there is one element in $N^-(R)$ that we have not yet used: the element $Q = R \setminus \{j\} \in N^-(R)$ and we may map this element to A to finish the definition of our injection π if needed. □

Claim 3.30. *For every $Q \in \mathcal{Q}$ and every $S \in N^-(Q)$, we have $d^+(Q) < d^+(S)$.*

Proof. The proof of this claim is similar to the proof of the previous claim. Let $Q \in \mathcal{Q}$ and $S \in N^-(Q)$. Once again we exhibit an injective function $\pi' : N^+(Q) \to N^+(S)$. We define $\pi'(R) = S \cup (R \setminus Q)$ for $R \in N^+(Q)$. Note that Q itself is never an image of π' thus a strict inequality holds. □

Claim 3.31.

$$\sum_{R \in \mathcal{R}} d^-(R) \geqslant \sum_{Q \in \mathcal{Q}} d^-(Q) \quad \text{and} \quad \sum_{Q \in \mathcal{Q}} d^+(Q) < \sum_{S \in \mathcal{S} \cup \{A\}} d^+(S).$$

Proof. We start by showing the first inequality. For $c \in \mathbb{N}$, let us define $\mathcal{R}_c = \{R \in \mathcal{R} \mid d^-(R) = c\}$ and distinguish two cases. Suppose first that there exists a $c \in \mathbb{N}$ such that $\mathcal{R}_c = \mathcal{R}$. Then we have $\sum_{R \in \mathcal{R}} d^-(R) = cm$. By Claim 3.29, $\forall Q \in \mathcal{Q}, d^-(Q) \leqslant c$ and therefore $\sum_{Q \in \mathcal{Q}} d^-(Q) \leqslant c(m-1) \leqslant \sum_{R \in \mathcal{R}} d^-(R)$.

Otherwise, $\mathcal{R}_c \neq \mathcal{R}$ for every choice of c. In this case, we define, for any integer $d < \max_{R \in \mathcal{R}} d^-(R)$, $\mathcal{R}_{\leqslant d} = \cup_{c \leqslant d} \mathcal{R}_c$ and remark that $\mathcal{R}_{\leqslant d} \neq \mathcal{R}$. Since we chose $(\mathcal{R}, \mathcal{S} \cup \{A\})$ to be minimal, Lemma 3.28 holds for the pair $(\mathcal{R}_{\leqslant d}, f(\mathcal{R}_{\leqslant d}))$. In particular, we can find a set $\mathcal{Q}_{\leqslant d}$ of size exactly $|\mathcal{R}_{\leqslant d}|$ such that $\mathcal{Q}_{\leqslant d} \subseteq \mathcal{Q}$ and every element in $\mathcal{Q}_{\leqslant d}$ lies on a path between an element of $\mathcal{R}_{\leqslant d}$ and an element of $f(\mathcal{R}_{\leqslant d})$. By definition, each $Q \in \mathcal{Q}_{\leqslant d}$ is in the in-neighbourhood of some $R \in \mathcal{R}_{\leqslant d}$, and therefore $d^-(Q) \leqslant d$ by Claim 3.29. We conclude that for any $d < \max_{R \in \mathcal{R}} d^-(R)$ there exist $|\mathcal{Q}_{\leqslant d}| = |\mathcal{R}_{\leqslant d}|$ vertices in $\mathcal{Q}$ of in degree at most d.

If we denote by $d_0 < d_1 < \cdots < d_k$ the in-degree sequence of $\mathcal{R}$, then the result of the last paragraph induces an injective function $\pi'' : \mathcal{R}_{\leqslant d_{k-1}} \to \mathcal{Q}$ as follows: we map $\mathcal{R}_{d_0}$ to $\mathcal{Q}_{\leqslant d_0}$, then map $\mathcal{R}_{d_1}$ to $\mathcal{Q}_{\leqslant d_1} \setminus \mathcal{Q}_{\leqslant d_0}$ and continue to map $\mathcal{R}_{d_i}$ to $\mathcal{Q}_{\leqslant d_i} \setminus \mathcal{Q}_{\leqslant d_{i-1}}$ for all $i \in [2, k-1]$. We argued in the previous paragraph that such injections exist. By construction, $\forall R \in \mathcal{R}_{\leqslant d_{k-1}},\ d^-(\pi''(R)) \leqslant d^-(R)$.

All vertices in $\mathcal{Q}$ are in the in-neighbourhood of some element of $\mathcal{R}$ and therefore $d^-(Q) \leqslant d_k$ for all $Q \in \mathcal{Q}$ by Claim 3.29. Since by assumption $|\mathcal{Q}| < |\mathcal{R}|$, this proves $\sum_{R\in\mathcal{R}} d^-(R) \geqslant \sum_{Q\in\mathcal{Q}} d^-(Q)$ since we can associate each term in the second sum to an element that is at least as large in the first sum (and all terms are non-negative).

The proof of the inequality $\sum_{Q\in\mathcal{Q}} d^+(Q) < \sum_{S\in\mathcal{S}\cup\{A\}} d^+(S)$ is analogous, but now the strict inequality follows from the strict inequality in Claim 3.30 instead of the weak inequality of Claim 3.29. □

We are now fully equipped to conclude the proof of Lemma 3.28. By double counting, $\sum_{Q\in\mathcal{Q}} d^-(Q) = \sum_{S\in\mathcal{S}\cup\{A\}} d^+(S)$ and $\sum_{R\in\mathcal{R}} d^-(R) = \sum_{Q\in\mathcal{Q}} d^+(Q)$. Using Claim 3.31 we deduce the following contradiction,

$$\sum_{Q\in\mathcal{Q}} d^-(Q) = \sum_{S\in\mathcal{S}\cup\{A\}} d^+(S) > \sum_{Q\in\mathcal{Q}} d^+(Q) = \sum_{R\in\mathcal{R}} d^-(R) \geqslant \sum_{Q\in\mathcal{Q}} d^-(Q).$$

This proves the lemma. □

Using Lemma 3.28 we can now prove the following special case of Lemma 3.27, which we will use to push through an inductive argument.

Lemma 3.32. *Let $3 \leqslant r \leqslant n$ be integers. Let $C^1, \ldots, C^{m-1}$ be skipless disjoint chains between the $(r-2)$th and the rth layers. Let $B \in \binom{[n]}{r}$ and let A be a subset of B of size at most $r-2$, such that $A, B \notin \cup_{i=1}^{m-1} C^i$.*

Then there exist m disjoint chains $D^1, \ldots, D^m$ with the following three properties.

- *For $i \in [m-1]$, the chain D^i starts in the $(r-2)$th layer, ends in the rth layer and is skipless.*
- *The chain D^m starts in A and intersects both the $(r-1)$th and the rth layer.*
- *The chains $D^1, \ldots, D^m$ cover the elements in $C^1, \ldots, C^{m-1}$ and A, B.*

Proof. We prove the claim by induction on m. The case $m = 1$ is immediate.

We let $\mathcal{R}, \mathcal{T}, \mathcal{S}$ denote the restriction of the chains to layers $r, r-1, r-2$ respectively, and add A to $\mathcal{S}$ and B to $\mathcal{R}$. That is,

$$\mathcal{R} = \left(\bigcup_i C^i \cap \binom{[n]}{r}\right) \cup \{B\},$$
$$\mathcal{T} = \bigcup_i C^i \cap \binom{[n]}{r-1},$$
$$\mathcal{S} = \left(\bigcup_i C^i \cap \binom{[n]}{r-2}\right) \cup \{A\}.$$

Let $\mathcal{Q}$ denote the set of all elements $Q \in \binom{[n]}{r-1}$ such that there exists $(R, S) \in \mathcal{R} \times \mathcal{S}$ satisfying $R \subseteq Q \subseteq S$. We define a bijection $f : \mathcal{R} \to \mathcal{S}$ with $f(B) = A$ and $f(X) \subseteq X$ for all $X \in \mathcal{R}$ using the given chains. Lemma 3.28 shows that $|\mathcal{Q}| \geqslant m$. Since $|\mathcal{T}| = m - 1$, $\mathcal{T}$ is a strict subset of $\mathcal{Q}$.

We consider the poset as a directed graph H' via an adjusted Hasse diagram as before: the vertex set consists of $V = \mathcal{R} \cup \mathcal{Q} \cup \mathcal{S}$, and $X \to Y$ is an arc in $E(H')$ if and only if $X \subsetneq Y$ and either $|Y| = |X| + 1$ or $X = A$ and $Y \in \mathcal{Q}$. Finding the desired chains $D^1, \ldots, D^m$, is equivalent to finding m vertex-disjoint paths between $\mathcal{R}$ and $\mathcal{S}$ in the induced subgraph $H_Q = H'[\mathcal{R} \cup \mathcal{T} \cup \mathcal{S} \cup \{Q\}]$ for some $Q \in \mathcal{Q}$. By Menger's theorem [142], there exist m vertex-disjoint paths if and only if there is no $(\mathcal{R}, \mathcal{S})$-cut of size $m - 1$, that is, there is no subset $\mathcal{C} \subseteq V$ with $|\mathcal{C}| = m - 1$ such that, for all pairs $(R, S) \in \mathcal{R} \times \mathcal{S}$, every path from R to S contains a vertex of $\mathcal{C}$.

Since $|\mathcal{T}| < |\mathcal{Q}|$, there is an element $Q_0 \in \mathcal{Q} \setminus \mathcal{T}$. By the discussion above, we may assume that an $(\mathcal{R}, \mathcal{S})$-cut $\mathcal{C}$ of size $m - 1$ exists in H_{Q_0}. We first show that $\mathcal{C} \not\subseteq \mathcal{Q}$. Indeed, for any $Q \in \mathcal{Q}$ there exists a pair $(R, S) \in \mathcal{R} \times \mathcal{S}$ such that $S \to Q \to R$ is a path in H'. When $\mathcal{C} \subseteq \mathcal{Q}$, all such paths in H_{Q_0} are cut off only when $\mathcal{C}$ contains all elements of $\mathcal{T} \cup \{Q_0\}$; but $|\mathcal{C}| = m - 1 < m = |\mathcal{T} \cup \{Q_0\}|$. So $\mathcal{C}$ must contain at least one element which is not in $\mathcal{Q}$.

We partition the size of the cut in three parts

$$m_1 = |\mathcal{R} \cap \mathcal{C}|, \quad m_2 = |\mathcal{Q} \cap \mathcal{C}|, \quad m_3 = |\mathcal{S} \cap \mathcal{C}|.$$

Consider the chains whose endpoints have not been touched by the cut. That is, let $\mathcal{R}^* \subseteq \mathcal{R}$ consist of the $R \in \mathcal{R}$ for which $R, f(R) \notin \mathcal{C}$, and let $\mathcal{S}^* = f(\mathcal{R}^*)$. Then $\mathcal{Q} \cap \mathcal{C}$ is an $(\mathcal{R}^*, \mathcal{S}^*)$-cut. Moreover,

$$m_2 = |\mathcal{Q} \cap \mathcal{C}| = (m - 1) - m_1 - m_3 < m - m_1 - m_3 \leqslant |\mathcal{R}^*|.$$

Let $\mathcal{T}^* \subseteq \mathcal{T}$ consist of the elements that lie on some chain C^i between $\mathcal{S}^*$ and $\mathcal{R}^*$. Since $\mathcal{Q} \cap \mathcal{C}$ is an $\mathcal{R}^*, \mathcal{S}^*$-cut of H_{Q_0}, it must in particular contain all elements of $\mathcal{T}^*$. Since $m_2 < |\mathcal{R}^*|$,

this means that $(A, B) \in (\mathcal{S}^* \times \mathcal{R}^*)$. Moreover, we may apply the induction hypothesis since $|\mathcal{R}^*| < |\mathcal{R}|$ (because $m_1 + m_3 > 0$). This gives us $|\mathcal{R}^*|$ chains which cover all elements in $\mathcal{T}^*$ and all intersect layer $r-1$, so in particular we obtain some element $Q_1 \in \mathcal{Q} \setminus \mathcal{T}^*$ such that there are $|\mathcal{R}^*| > m_2$ vertex-disjoint $\mathcal{S}^* - \mathcal{R}^*$ paths in $H^* = H'[\mathcal{R}^* \cup \mathcal{T}^* \cup \{Q_0, Q_1\} \cup \mathcal{S}^*]$. We distinguish two cases.

- Suppose that $Q_1 \notin \mathcal{T}$. In this case we have obtained our desired chain decomposition. Indeed, we keep the chains between $\mathcal{S} \setminus \mathcal{S}^*$ and $\mathcal{R} \setminus \mathcal{R}^*$ as they are and since $\mathcal{T}^* \cup \{Q_1\}$ is disjoint from those chains, we may use the $|\mathcal{R}^*|$ chains between $\mathcal{R}^*$ and $\mathcal{S}^*$ that we obtained by induction in order to define the remaining chains.

- Suppose that $Q_1 \in \mathcal{T}$. In that case, H^* is an induced subgraph of H_{Q_0}. This gives a contradiction: H^* has $|\mathcal{R}^*| > m_2$ vertex disjoint paths between $\mathcal{R}^*$ and $\mathcal{S}^*$, whereas $\mathcal{Q} \cap \mathcal{C}$ gives an $(\mathcal{R}^*, \mathcal{S}^*)$-cut of size m_2 in H_{Q_0}. □

From this, we will deduce the case of general s.

Proof of Lemma 3.27. We prove the lemma by induction on m. The case $m = 1$ is immediate. Suppose the claim has been shown for all $m' < m$.

Let $C^1, \dots, C^m$ be the given chain decomposition, where C^m starts in $A \in \binom{[n]}{\leqslant s}$ and ends in $B \in \binom{[n]}{r}$, and the first $m-1$ chains are between layers s and r. Let $t \in [s+1, r]$. We say the chains $D^1, \dots, D^m$ are t-good if the first $m-1$ chains are skipless and between layers s and r, D^m is between A and B and intersects layers $t, \dots, r$, and $\cup_{i=1}^m C^i \subseteq \cup_{i=1}^m D^i$.

We first argue that there exists an r-good decomposition. Indeed, applying the induction hypothesis to the first $m' = m-1$ chains, we can find chains $D^1, \dots, D^{m-1}$ between layers s and r that are skipless and such that $\cup_{i=1}^{m-1} C^i \subseteq \cup_{i=1}^{m-1} D^i$. By removing the elements from C^m that also appear in some D^i, we have obtained an r-good decomposition for $C^1, \dots, C^m$.

Let $t \leqslant r$ be minimal for which a t-good decomposition $D^1, \dots, D^m$ exists. Suppose towards a contradiction that $t > s+1$. Let B' be the element of D^m in layer t. Since $t > s+1$, we find $t-2 \geqslant s$ and so the chains $D^1, \dots, D^{m-1}$ all intersect layer $t-2$. We can apply Lemma 3.32 on the chains $D^1, \dots, D^{m-1}$ restricted to layers $s' = t-2$ and $r' = t$, and elements A and B'. This produces a set $\mathcal{C}_1$ of chains. Let $\mathcal{C}_0$ and $\mathcal{C}_2$ be the restrictions of $D^1, \dots, D^m$ to layers $s, \dots, t-2$ and to layers $t, \dots, r$ respectively. Then each chain of $\mathcal{C}_1$ shares a vertex with exactly one chain of $\mathcal{C}_0$ and exactly one chain of $\mathcal{C}_2$. Hence, there is only one way to merge these chains in a chain decomposition $E^1, \dots, E^m$. This chain decomposition is $(t-1)$-good, contradicting the minimality of t. Therefore, there exists an $(s+1)$-good decomposition $D^1, \dots, D^m$, as claimed by the lemma. □

We will obtain Theorem 3.10 as a corollary of the following lemma. The lemma is stated in the way that we wish to apply it in the proof of Theorem 3.8.

Lemma 3.33. *Let $\mathcal{F} \subsetneq 2^{[n]}$ be k-antichain saturated. Then $\mathcal{F}$ has a chain decomposition into $k-1$ skipless chains.*

Proof. Suppose, towards a contradiction, that $\mathcal{F}$ has no chain decomposition $C^1, \dots, C^{k-1}$ for which the first $i+1$ chains are skipless, but it does have one for which the first i are skipless. Note that, we can always rearrange the chain decomposition such that C^1 is skipless, else $\mathcal{F}$ would not be saturated. This means we have $1 \leqslant i < k-1$.

Amongst the decompositions for which the first i chains are skipless, we choose a decomposition $C^1, \dots, C^{k-1}$ which minimises the "number of layers the $(i+1)$th chain skips". That is, the decomposition which minimises

$$\max_{X \in C^{i+1}} |X| - \min_{Y \in C^{i+1}} |Y| + 1 - |C^{i+1}|.$$

By assumption, we can find $A \subseteq B$ consecutive in C^{i+1} with $|B| > |A| + 1$ such that C^{i+1} is skipless between B and its maximal element. After renumbering, we can assume that for some $j \in [0, i]$, the chains $C^1, \dots, C^j$ have elements present on layers $|B|-2, |B|-1$ and $|B|$, whereas $C^{j+1}, \dots, C^i$ miss an element either on layer $|B|-2$ or on layer $|B|$. (Here we use that $C^1, \dots, C^i$ are skipless.) In particular, if C^a where $a \in [j+1, i]$ has an element on layer $|B|-1$, then it is its minimal or maximal element, and so we can move it to another chain without creating any skips in the chain C^a.

We apply Lemma 3.27 to the chains $C^1, \dots, C^j$ restricted to layers $|B|-2, |B|-1, |B|$, and $A \subseteq B$ to obtain disjoint chains $D^1, \dots, D^{j+1}$ with the following properties:

- $\cup_{a=1}^{j} C^a \cup \{A, B\} \subseteq \cup_{a=1}^{j+1} D^a$;
- $D^1, \dots, D^j$ are skipless, start in layer $|B|-2$ and end in layer $|B|$;
- D^{j+1} contains A and elements on layers $|B|-1$ and $|B|$.

Since the chains $C^1, \dots, C^j$ have an element on layer $|B|-1$, there is a unique $X \in \cup_{a=1}^{j+1} D^a$ with $|X| = |B|-1$ such that $X \notin \cup_{a=1}^{j} C^a$.

The chains $D^1, \dots, D^{j+1}$ define a matching M between layer $|B|$ and layer $|B|-1$ of size $j+1$. We will use this to reroute the chains into a "better" chain decomposition and arrive at a contradiction. A possible configuration is depicted in Figure 3.6. We define the chain decomposition $E^1, \dots, E^{k-1}$ as follows.

For $a \in [j]$, let $b \in [j]$ be such that D^b contains the unique element in C^a of size $|B|-2$. Let $a' \in [j] \cup \{i+1\}$ be the index such that D^b contains the unique element in $C^{a'}$ of size $|B|$. We set

$$E^a = \left[C^a \cap \binom{[n]}{\leqslant |B|-2} \right] \cup D^b \cup \left[C^{a'} \cap \binom{[n]}{\geqslant |B|} \right].$$

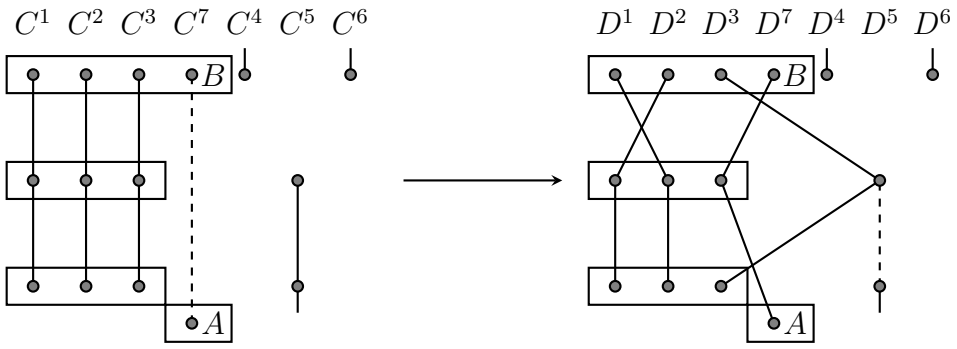


Figure 3.7: An example of a possible rearrangement as done in the proof of Lemma 3.33 (for $j = 3$ and $i = 6$). The sets A and B are part of the chain C_7.

Note that our assumption that $C^1, \dots, C^{i+1}$ are skipless from layer $|B|$ upwards means the chain E^a must be skipless as well.

For $a \in [j+1, i]$, we let $E^a = C^a \setminus \{X\}$. Either we kept the chain the same, or we removed the minimal or maximal element, so these chains are also skipless. For $a \in [i+2, n]$, we also set $E^a = C^a \setminus \{X\}$.

For $a = i + 1$, let $C^{a'}$ be the unique chain which contains the element of D^{j+1} of size $|B|$. We set

$$E^{i+1} = \left[C^{i+1} \cap \binom{[n]}{\leqslant |A|}\right] \cup D^{j+1} \cup \left[C^{a'} \cap \binom{[n]}{\geqslant |B|}\right].$$

The chains $E^1, \dots, E^{k-1}$ form a chain decomposition of $\mathcal{F} \cup \{X\}$ (which must equal $\mathcal{F}$ in this case because $\mathcal{F}$ is k-antichain saturated). The chains $E^1, \dots, E^i$ are skipless and the chain E^{i+1} skips one fewer layers than the chain C^{i+1}, contradicting the optimality of $C^1, \dots, C^{k-1}$. □

We recall the statement of Theorem 3.10: if $\mathcal{F}$ admits a chain decomposition into m chains, then it can be covered by m skipless chains.

Proof of Theorem 3.10. By assumption, $\mathcal{F}$ does not contain an antichain of size $m + 1$. Let $\mathcal{F}'$ be obtained from $\mathcal{F}$ by greedily adding sets until the set system has become $(m + 1)$-antichain saturated. If $\mathcal{F}' = 2^{[n]}$, then we find a skipless chain decomposition for $\mathcal{F}'$ by Lemma 0.3. Otherwise, we can find a chain decomposition for $\mathcal{F}'$ into $m + 1 - 1 = m$ skipless chains by Lemma 3.33. These chains cover $\mathcal{F}$ as desired. □

3.4.2 Overview of the proof of Theorem 3.8

This section focuses on Theorem 3.8. The complete proof is quite involved, as it requires improving both the previously known lower bound and the upper bound. In the following, we first prove the lower bound part of the statement. Second, we provide a proof for a simpler subcase of the general upper bound construction of Theorem 3.8. This simpler construction serves as the building block for our more general upper bound construction. We conclude this section by describing our general construction without including its correctness proof, which is available in [25].

Lower bound of Theorem 3.8

Let us first recall the set-up. Given a natural number k, let ℓ be the smallest integer j such that $\binom{j}{\lfloor j/2 \rfloor} \geqslant k-1$. We may assume that $n \geqslant \ell$. Let $\mathcal{C}(m,t)$ denote the initial segment of layer t of size m under the colexicographic order. Let $c_t = k-1$ for all $t \in [\lfloor \ell/2 \rfloor, \lfloor n/2 \rfloor]$. For $0 \leqslant t < \lfloor \ell/2 \rfloor$, we define $c_t = \nu\left(\mathcal{C}(c_{t+1}, t+1)\right)$. (Recall that $\nu(\mathcal{B})$ is the size of the largest of a matching between $\mathcal{B}$ and its shadow, as defined in Section 3.2.)

The lower bound of Theorem 3.8 follows directly from the lemma below, since the desired lower bound for the upper layers follows by symmetry.

Lemma 3.34. *For any k-antichain saturated set system $\mathcal{F} \subsetneq 2^{[n]}$, $|\mathcal{F}_t| \geqslant c_t$ for any $t \leqslant \lfloor n/2 \rfloor$.*

Proof. Suppose that $\mathcal{F} \subsetneq 2^{[n]}$ is k-antichain saturated. By Lemma 3.33, there is a skipless chain decomposition $C^1, \ldots, C^{k-1}$ for $\mathcal{F}$. Let $\mathcal{F}_t = \mathcal{F} \cap \binom{[n]}{t}$. We define $\mathcal{D}(\mathcal{F}_t)$ as the sets $A \in \mathcal{F}_{t-1}$ for which the chain C^i that contains A also contains an element of $\mathcal{F}_t$. The following claim is key to our proof.

Claim 3.35. *For all $t \in [n]$, $|\mathcal{D}(\mathcal{F}_t)| = \nu(\mathcal{F}_t)$.*

Proof. By definition, there is a matching from $\mathcal{F}_t$ to $\mathcal{D}(\mathcal{F}_t) \subseteq \partial\mathcal{F}_t$ of size $|\mathcal{D}(\mathcal{F}_t)|$, and hence, $\nu(\mathcal{F}_t) \geqslant |\mathcal{D}(\mathcal{F}_t)|$. We now focus on the opposite inequality.

Suppose, towards a contradiction, that there is a t for which $|\mathcal{D}(\mathcal{F}_t)| < \nu(\mathcal{F}_t)$. Let M be a matching between $\mathcal{F}_t$ and $\partial\mathcal{F}_t$ of size $\nu(\mathcal{F}_t)$, and let M' be the matching between $\mathcal{F}_t$ and $\mathcal{C}(\mathcal{F}_t)$ corresponding to the inclusions in the chains (i.e. X is matched to Y if X and Y are in the same chain). Consider the multigraph where the vertices are $\binom{[n]}{t} \cup \binom{[n]}{t-1}$ and the edge set is $M \cup M'$. The non-empty components of this graph are paths and even cycles which alternate between edges from M and M' (with no multiedges), and multiedges which have one edge from M and one edge from M'. Since $|M| > |M'|$ there must be some component P which is a path that starts and ends with edges from M. We will reroute some of the chains so that they use the edges from M instead of the edges from M', increasing the size of $\mathcal{D}(\mathcal{F}_t)$.

If a chain C^a is not incident with an edge in this path, let $D^a = C^a$ (i.e. the chain is unchanged). One end of P must be in layer t and one end in layer $t-1$, and we order the edges starting from the end in layer t. If $e \in M$ is not the last edge in the path, then it connects a set $X \in C^a$ of size t to a set $Y \in C^b$ of size $t-1$, and we replace C^a by

$$D^a = \left(C^a \cap \binom{[n]}{\geqslant t}\right) \cup \left(C^b \cap \binom{[n]}{\leqslant t-1}\right).$$

If $e \in M$ is the last edge in the path, there are two cases. The edge may connect a set $X \in C^a$ of size t to a set Y of size $t-1$ which is not in any other chain, in which case we set $D^a = \left(C^a \cap \binom{[n]}{\geqslant t}\right) \cup \{Y\}$. Then $D^1, \dots, D^{k-1}$ gives a decomposition of $\mathcal{F} \cup \{Y\}$ into $k-1$ chains and this contradicts the assumption that $\mathcal{F}$ is k-antichain saturated. The other case is where the edge connects a set $X \in C^a$ of size t to a set $Y \in C^b$ of size $t-1$. Since there is no edge in M' incident to Y, it must be the largest set in C^b. In this case, we set $D^a = \left(C^a \cap \binom{[n]}{\geqslant t}\right) \cup C^b$. The $k-2$ chains $D^1, \dots, D^{b-1}, D^{b+1}, \dots, D^{k-1}$ now cover all the elements of $\mathcal{F}$ and we may still define the chain D^b freely. We can choose any set A which is not in $\mathcal{F}$ and set $D^b = \{A\}$. Then $D^1, \dots D^{k-1}$ is a chain decomposition of $\mathcal{F} \cup \{A\}$ into $k-1$ chains, a contradiction. □

Lemma 0.3 shows that, for all $t > \lfloor n/2 \rfloor$ there is a matching between $\mathcal{F}_t$ and $\partial\mathcal{F}_t$ of size $|\mathcal{F}_t|$, which implies $\nu(\mathcal{F}_t) = |\mathcal{F}_t|$. Using the claim above, every chain with a set in layer t must have a set in layer $t-1$ for all $t > \lfloor n/2 \rfloor$. The set system $\overline{\mathcal{F}} = \{[n] \setminus F : F \in \mathcal{F}\}$ is also k-antichain saturated. Applying Claim 3.35 to $\overline{\mathcal{F}}$, we find that every chain with a set of size $s < \lceil n/2 \rceil$ must have a set of size $s+1$ as well. Putting these together gives the following claim.

Claim 3.36. *For all $i \in [k-1]$, C^i contains an element from layer $\lfloor n/2 \rfloor$.*

An immediate consequence of Claim 3.35 is that $|\mathcal{F}_{t-1}| \geqslant \nu(\mathcal{F}_t)$. Together with Lemma 3.21, this shows

$$|\mathcal{F}_{t-1}| \geqslant \nu(\mathcal{F}_t) \geqslant \nu(\mathcal{C}), \tag{3.2}$$

where $\mathcal{C}$ is an initial segment of colex on $\binom{[n]}{t}$ of size $|\mathcal{F}_t|$. We already have $|\mathcal{F}_{\lfloor n/2 \rfloor}| = k-1$ (by Claim 3.36) and we want this for $\mathcal{F}_t$ down to $t = \lfloor \ell/2 \rfloor$. From (3.2), we can push this downwards at least when $\nu(\mathcal{C}) = |\mathcal{C}|$, and the following claim shows that this holds for all $t > \lfloor \ell/2 \rfloor$.

Claim 3.37. *For $t > \lfloor \ell/2 \rfloor$, an initial segment of colex $\mathcal{C}$ on layer t of size at most $k-1$ has $\nu(\mathcal{C}) = |\mathcal{C}|$, and so $|\mathcal{F}_{t-1}| \geqslant |\mathcal{F}_t|$.*

Proof. Let ℓ^* be the largest element in any set in $\mathcal{C}$ i.e. $\ell^* = \max(\bigcup_{A\in\mathcal{C}} A)$. If $t > \lfloor \ell^*/2 \rfloor$, then applying Lemma 0.3 to $[\ell^*]$ shows that there is a matching from $|\mathcal{C}|$ to layer $t-1$ of $[\ell^*]$ of size $|\mathcal{C}|$, and we are done. Suppose instead that $t \leqslant \lfloor \ell^*/2 \rfloor$. Since $\mathcal{C}$ is an initial segment of colex, it must contain all subsets of $[\ell^*-1]$ of size t as well as a set containing ℓ^*, but this means $\mathcal{C}$ contains too many sets. Indeed,

$$1 + \binom{\ell^*-1}{t} \geqslant 1 + \binom{2t-1}{t} \geqslant 1 + \binom{\ell}{\lfloor \ell/2 \rfloor} \geqslant k. \qquad \square$$

Combined with Claim 3.36, we find that layers $\lfloor \ell/2 \rfloor$ up to $\lfloor n/2 \rfloor$ all contain $k-1$ elements of $\mathcal{F}$.

For $t < \lfloor \ell/2 \rfloor$, if $|\mathcal{F}_{t+1}| \geqslant c_{t+1}$ then (3.2) shows that

$$|\mathcal{F}_t| \geqslant \nu(\mathcal{C}(|\mathcal{F}_{t+1}|, t+1)) \geqslant \nu(\mathcal{C}(c_{t+1}, t+1)) = c_t,$$

which concludes the proof of Lemma 3.34. $\square$

Note that, by complementing every set, Lemma 3.34 can be used to deduce lower bounds for layers close to n. The lower bound of Theorem 3.8 follows from using this observation and summing over every layer.

Let us remark that for infinitely many values of k, a matching upper bound to Theorem 3.8 was already known [73] (this upper bound construction is briefly explained below) which works for all $n \geqslant \ell + 1$. Therefore Lemma 3.34 combined with the simple observation that $\nu\left(\binom{[m]}{r}\right) = \binom{m}{r-1}$ provided $r \leqslant \lceil m/2 \rceil$, gives the following corollary.

Corollary 3.38. *Let ℓ, k, n be integers such that $\binom{\ell}{\lfloor \ell/2 \rfloor} = k-1$. If $n \leqslant \ell$ then $\mathrm{sat}^*(n,k) = 2^n$. If $n \geqslant \ell+1$, then*

$$\mathrm{sat}^*(n,k) = 2\sum_{j=0}^{\lfloor \ell/2 \rfloor} \binom{\ell}{j} + (k-1)(n-1-2\lfloor \ell/2 \rfloor).$$

When $k-1 = \binom{\ell}{\lfloor \ell/2 \rfloor}$, we remark that all minimal k-antichain saturated set systems have a similar shape: layer $\lfloor \ell/2 \rfloor$ is the lowest layer with $k-1$ elements and induces an isomorphic copy of colex, layer $n - \lfloor \ell/2 \rfloor$ is the highest layer with $k-1$ elements and contains the complements of an isomorphic copy of an initial segment of colex, and the elements in between these two layers can be covered by $k-1$ skipless chains.

A simpler upper bound (Corollary 3.38)

Let us first describe the upper bound construction mentioned above, studied in [73]. Let ℓ, k, n be integers such that $\binom{\ell}{\lfloor \ell/2 \rfloor} = k-1$ and $n \geqslant \ell + 1$. We show,

$$\mathrm{sat}^*(n,k) \leqslant 2\sum_{j=0}^{\lfloor \ell/2 \rfloor} \binom{\ell}{j} + (k-1)(n-1-2\lfloor \ell/2 \rfloor). \tag{3.3}$$

We define a set system $\mathcal{F} \subseteq 2^{[n]}$ that is k-antichain saturated.

For $t \leqslant \lfloor \ell/2 \rfloor$, the sets of size t in $\mathcal{F}$ are exactly the subsets of $[\ell]$ of size t, and for $t \geqslant n - \lfloor \ell/2 \rfloor$, we add to $\mathcal{F}$ all subsets $X \subseteq [n]$ of size t for which $[n] \setminus X$ is a subset of $[\ell]$. There are $k-1$ sets $\mathcal{F}$ of size $\lfloor \ell/2 \rfloor$ and $n - \lfloor \ell/2 \rfloor$, and we will join these up using Theorem 3.10.

For ℓ odd, we first fix a matching M between $\binom{[\ell]}{\lfloor \ell/2 \rfloor}$ and $\binom{[\ell]}{\lceil \ell/2 \rceil}$, which exists by Lemma 0.3. When ℓ is even, we let M be the identity. We denote by $M(X)$ the element matched to X by M. Let $f : \mathcal{F}_{\lfloor \ell/2 \rfloor} \to \mathcal{F}_{n-\lfloor \ell/2 \rfloor}$ be given by

$$f(X) = M(X) \cup [\ell+1, n],$$

and note that $X \subseteq f(X)$ for all $X \in \mathcal{F}_{\lfloor \ell/2 \rfloor}$. To complete the family $\mathcal{F}$, we take any set of $k-1$ disjoint skipless chains between $\mathcal{F}_{\lfloor \ell/2 \rfloor}$ and $\mathcal{F}_{n-\lfloor \ell/2 \rfloor}$, which we know exist by Theorem 3.10.

To see that $\mathcal{F}$ has no antichain of size k, we note that it allows a decomposition into $k-1$ chains. Indeed, we may extend the previously obtained $k-1$ chains between layers $\lfloor \ell/2 \rfloor$ and $n - \lfloor \ell/2 \rfloor$, using any chain decomposition of $[\ell]$ restricted to the lowest $\lfloor \ell/2 \rfloor$ layers. We can similarly extend the chains to the layers $n - \lfloor \ell/2 \rfloor + 1, \ldots, n$.

To see that $\mathcal{F}$ is saturated, note that we clearly cannot add any subset of size $t \in [\lfloor \ell/2 \rfloor, n - \lfloor \ell/2 \rfloor]$ since $\mathcal{F}$ already contains $k-1$ subsets of size t. For $t < \lfloor \ell/2 \rfloor$, any subset of size t that is not yet in $\mathcal{F}$ must contain some element $i > \ell$ and is therefore incomparable to the $k-1$ elements of $\mathcal{F} \cap \binom{[n]}{\lfloor \ell/2 \rfloor}$. A similar argument holds for $t > n - \lfloor \ell/2 \rfloor$.

By counting the number of sets in each layer, we find that

$$|\mathcal{F}| = 2\sum_{j=0}^{\lfloor \ell/2 \rfloor} \binom{\ell}{j} + (k-1)(n-1-2\lfloor \ell/2 \rfloor),$$

as required.

General upper bound (Theorem 3.8)

Let us now describe our general upper bound construction. Only the construction is explained here, the proof that it is indeed a A_k-saturated family is available in [28].

Recall that Lemma 3.34 tells us that the set system $\mathcal{F}$ that we wish to construct satisfy $\nu(\mathcal{F}_t) = \nu(\mathcal{C}(|\mathcal{F}_t|, t))$ for all $t \leqslant \lfloor \ell/2 \rfloor$ while maintaining that $\mathcal{F}$ can be covered by $k-1$ chains.

Suppose that each set in $\mathcal{C}(m, r)$ is in a chain and consider how many continue to the layer below. If we consider the r-cascade notation of m (defined in Section 3.2), Lemma 3.22 shows that the difference $m - \nu(\mathcal{C}(m, r))$ comes only from an initial sequence and that, for remaining binomial terms, we always need to continue the chains to the layer below. This gives us some freedom to not use the initial segment of colex for these continuing chains, and we will instead take elements that ensure that we will able to use a suitable initial segment of colex on smaller layers. To do this, we will change the later terms in the r-cascade notation of m, so that they are part of the expansion of a later layer and use the new expansion to define $\mathcal{F}$. We now introduce this different way of writing of m as a sum of binomial coefficients that gives our new expansion.

Given $m, r \geqslant 1$ such that $m \geqslant \binom{2r-1}{r}$. Let the *$r$-expansion* of m be

$$m = \binom{a_{r_1}}{r_1} + \cdots + \binom{a_{r_s}}{r_s}$$

recursively formed as follows. Let $r_1 = r$ and define a_{r_1} as the maximum j such that $\binom{j}{r_1} \leqslant m$. Note that $a_{r_1} \geqslant 2r_1 - 1$. Set $m' = m - \binom{a_{r_1}}{r_1}$. If $m' = 0$, we are done. Otherwise, let r' be the maximum $j \leqslant r-1$ such that $\binom{2j-1}{j} \leqslant m'$ and form the r-expansion of m by appending to $\binom{a_{r_1}}{r_1}$ the r'-expansion of m'. It is easy to see that this is well-defined and must terminate.

As an example, let us consider the 5-expansion of $m = 1011$. Since $\binom{12}{5} < 1011 < \binom{13}{5}$, we take $a_{r_1} = 12$ (and $r_1 = 5$). This means $m' = 219$, and the largest integer $j \leqslant 4$ such that $\binom{2j-1}{j} \geqslant m'$ is $j = 4$ (we also have $\binom{9}{5} \leqslant m'$, but this is not allowed). We therefore append the 4-expansion of 219. Calculating this recursively in the same manner, we see $a_{r_2} = 10$ (and $r_2 = 4$), which leaves a remainder of 9. Since $\binom{4}{2} \leqslant 9 < \binom{5}{3}$, we append the 2-expansion of 9, which is $\binom{4}{2} + \binom{3}{1}$. This gives the 5 expansion of 1011 as

$$1011 = \binom{12}{5} + \binom{10}{4} + \binom{4}{2} + \binom{3}{1}.$$

Construction for Theorem 3.8. Let ℓ be the smallest integer j such that $\binom{j}{\lfloor j/2 \rfloor} \geqslant k-1$. Let

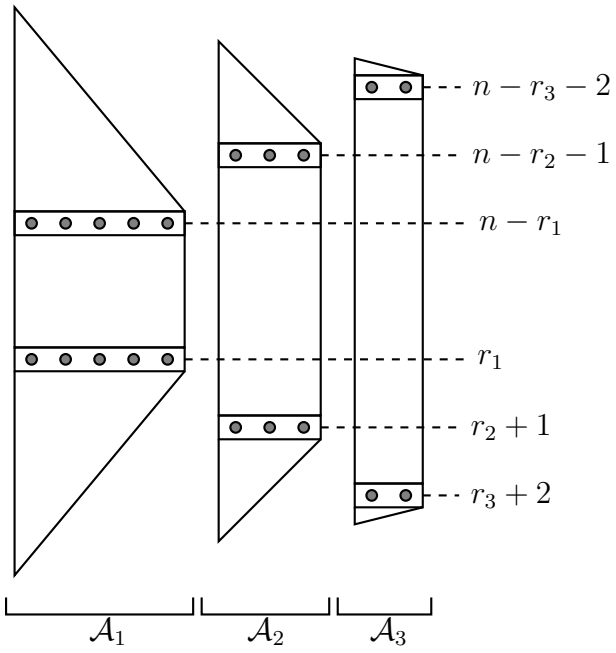


Figure 3.8: The shape of our upper bound construction is depicted. We represent layers horizontally (starting with the lowest at the bottom). For some, we indicated the number of the layer (e.g. layer $n - r_3 - 2$).

the $\lfloor \ell/2 \rfloor$-expansion of $k-1$ be

$$k-1 = \binom{a_{r_1}}{r_1} + \cdots + \binom{a_{r_s}}{r_s}$$

Through some omited proof, we can guarantee that $\lfloor \ell/2 \rfloor = r_1 > \cdots > r_s \geqslant 1$, $a_{r_1} > \cdots > a_{r_s} > 0$ and $r_i \leqslant \lceil a_{r_i}/2 \rceil$ for all $i \in [s]$. Note that the specific case where $a_{r_1} = \ell$ has been treated in the previous subsection as the upper bound construction for Corollary 3.38, so we can suppose that $a_{r_1} \leqslant \ell - 1$.

We now define our construction by processing each of the terms in this expansion. Initialise $\mathcal{I}$ as an empty set of chains. For each $i \in [s]$, let $\mathcal{A}_i$ be the set system consisting of sets of the form

$$A = X \cup \{a_{r_1}+1, a_{r_2}+1, \ldots, a_{r_{i-1}}+1\}$$

where X is a subset of $[a_{r_i}]$ of size at most r_i. Note that the largest element in any of these sets is either a_{r_1} or $a_{r_1}+1$, and hence all sets are contained in $[\ell]$.

Since $r_i \leqslant \lceil a_{r_i}/2 \rceil$, we can cover $\mathcal{A}_i$ with $\binom{a_{r_i}}{r_i}$ disjoint chains, and we add these chains to our collection of chains $\mathcal{I}$. Indeed, we may start with the chains from a symmetric chain decomposition of $2^{[a_{r_i}]}$ and add the elements $a_{r_1}+1, a_{r_2}+1, \ldots, a_{r_{i-1}}+1$ to every set. Then we discard any sets which are not in $\mathcal{A}_i$ and remove any empty chains to leave the required chain covering.

Define $f : 2^{[n]} \to 2^{[n]}$ by $f(A) = \{i \in [n] : n+1-i \notin A\}$. Form a second set of chains $\mathcal{I}'$ by replacing each chain $C \in \mathcal{I}$ by $\{f(A) : A \in C\}$. Since we have assumed that $n \geqslant 2\ell+1$, we have that $A \subseteq f(A)$ for any set $A \subseteq [\ell]$, and these are indeed chains. The chains in $\mathcal{I}'$ are also disjoint and we can apply Theorem 3.10 to find disjoint chains $D^1, \ldots, D^{k-1}$ which cover all sets in $\mathcal{I} \cup \mathcal{I}'$ and are skipless. We take $\mathcal{F}$ to be the union of $D^1, \ldots, D^{k-1}$. See Fig. 3.8 for a depiction of a set system constructed as above. We only note here, that the set system $\mathcal{F}$ is the union of $k-1$ chains, so cannot contain an antichain of size k. The fact that $\mathcal{F}$ is indeed k saturated is proven in [28].

□

3.5 Conclusion and perspectives

Despite extensive efforts from researchers, our understanding of the possible behaviours of the parameter sat^* remains limited. By combining the results of [77] and Theorem 3.15, we obtain the following theorem, which summarises our knowledge of the possible range of values taken by $\mathrm{sat}^*(n, P)$.

Theorem 3.39. *For any poset P and any $n \in \mathbb{N}$, either*

$$\mathrm{sat}^*(n, P) = O(1) \quad \textit{or} \quad \Omega(\sqrt{n}) = \mathrm{sat}^*(n, P) = O(n^{|P|}).$$

To this day, only two types of behaviour are known: either constant or linear. Moreover, examples of both behaviours have been known for quite some time, such as $\mathrm{sat}^*(n, C_2) = 1$ and $\mathrm{sat}^*(n, A_2) = n + 1$. The primary conjecture in the field can be stated as follows:

Conjecture 3.40. *For any poset P and any $n \in \mathbb{N}$, either*

$$\mathrm{sat}^*(n, P) = O(1) \quad \textit{or} \quad \mathrm{sat}^*(n, P) = \Theta_P(n).$$

A weaker version of this conjecture was first formulated in [116], where $\Theta_P(n)$ was replaced by $\Omega(n)$. If we believe Conjecture 3.40 to be true, then we can speculate on possible structural explanations for the linear behaviour. Perhaps, there always exists a saturated family using only a constant number of elements on each layer of the hypercube.

Conjecture 3.41. *For any poset P, and any $n \in \mathbb{N}$, there exists a P-saturated family $\mathcal{F}$ in $2^{[n]}$ such that,*

$$|\mathcal{F} \cap \binom{[n]}{t}| = O(1).$$

To understand the possible behaviour, it might be a good idea to first focus on the simplest case and attempt to deepen our understanding of it. Almost nothing is known about posets P satisfying $\mathrm{sat}^*(n, P) = O(1)$. From a theoretical perspective, developing necessary and/or sufficient conditions for a poset to have a bounded saturation number would be valuable. For example, finding a characterisation of posets P such that $\mathrm{sat}^*(n, P) = O(1)$. From an algorithmic perspective, the following problem remains widely open.

Problem 3.42. *What is the complexity class of the following algorithmic problem:*
Input: *A poset P*
Output: YES *if* $\mathrm{sat}^*(n, P) = O(1)$ *and* NO *otherwise.*

We remark that, to our knowledge, the problem discussed above is not even known to be decidable.

On the contrary, if we believe Conjecture 3.40 to be false, we would need to exhibit a poset P for which $\mathrm{sat}^*(n, P) = \omega(1)$ but $\mathrm{sat}^*(n, P) \neq \Theta(n)$. Despite multiple papers focusing on the diamond poset $\Diamond$ [73, 95], it is still a possible candidate for disproving Conjecture 3.40. The following conjecture was proposed by Ivan in [95].

Conjecture 3.43. *For any $n \in \mathbb{N}$, $\mathrm{sat}^*(n, \Diamond) = \Theta(\sqrt{n})$.*

Chapter 4

Spread embedding in graphs of high minimum degree

This chapter presents results about how to construct "uniform-like" random embeddings, called spread embeddings *in dense graphs. This is joint work with Clément Legrand-Duchesne and Alp Müyesser [30].*

4.1 Introduction

There is a large body of results in extremal graph theory focusing on determining the minimum degree threshold which forces the containment of a target subgraph. For example, a classical result of Dirac [62] states that any n-vertex graph with minimum degree at least $n/2$ contains a Hamilton cycle. Although this result is tight, graphs with minimum degree $n/2$ are quite dense, so it is natural to suspect that they are Hamiltonian in a rich sense. In this direction, Sárközy, Selkow, and Szemerédi [165] showed that n-vertex graphs with minimum degree $n/2$ contain $\Omega(n)^n$ distinct Hamilton cycles (we refer to such results as *enumeration* results, see also [58]). Moreover, randomly sparsifying the edge set of an n-vertex graph with minimum degree $n/2$ yields, with high probability, another Hamiltonian graph, as long as each edge is kept with probability $\Omega(\log n/n)$. This follows from an influential result of Krivelevich, Lee, and Sudakov [125] (such results are referred to as *robustness* results, see [174]), which generalises Pósa's celebrated result stating that the random graph $G(n, C\log n/n)$ is Hamiltonian with high probability.

The study of random graphs was recently revolutionised by Frankston, Kahn, Narayanan, and Park's [76] proof of the *fractional expectation threshold vs. threshold* conjecture of Tala-

grand [175] (see also [150] for a proof of the even stronger Kahn–Kalai conjecture [103]). In our context, these breakthroughs imply that the enumeration and robustness results stated in the first paragraph, which themselves are fairly general, admit a a further common generalisation. To state this generalisation, we need the language of *spread distributions* which we will define momentarily. In a nutshell, the key idea is to show that a graph G, with minimum degree large enough to necessarily contain a copy of a target graph H, actually supports a *random embedding* of H that (roughly speaking) resembles a uniformly random function from $V(H)$ to $V(G)$. The formal definition we use is below.

Definition 4.1 ([154]). *Let X and Y be finite sets, and let μ be a probability distribution over injections $\varphi : X \to Y$. For $q \in [0,1]$, we say that μ is q-spread if for every two sequences of distinct vertices $x_1, \dots, x_s \in X$ and $y_1, \dots, y_s \in Y$,*

$$\mu\left(\{\varphi : \varphi(x_i) = y_i \text{ for all } i \in [s]\}\right) \leqslant q^s.$$

In our context, $X = V(H)$, $Y = V(G)$, $|X| = |Y| = n$, and μ is a probability distribution over embeddings of H into G. The gold standard for us is constructing distributions μ that are $O(1/n)$-spread. Such distributions have the optimal spread (up to the value of the implied constant factor) that is also attained by a random injection from $V(H)$ to $V(G)$. We remark that Definition 4.1, originally introduced in [154], is different to the usual definition of spreadness phrased in terms of edges, instead of vertices. However, for embedding spanning subgraphs, the above definition turns out to be more convenient (see [115, 154] for more details).

The breakthroughs on the Kahn-Kalai conjecture have created a lot of incentive to show "spread versions" of Dirac-type results in graphs and hypergraphs, as such results directly imply enumeration and robustness results, thereby coalescing two streams of research which have, until now, been investigated independently. We refer the reader to the recent papers [154, 115, 105, 100, 14] that obtain several results in this direction (see also [7]). Most of the aforementioned work focuses on constructing spread distributions for target graphs with rather simple structures, such as perfect matchings or Hamilton cycles. One notable exception is the result from [154] for bounded degree trees. To introduce this result, we first cite the following classical result in extremal graph theory.

Theorem 4.2 (Komlós–Sárközy–Szemerédi [121]). *For every $\Delta \in \mathbb{N}$ and $\alpha > 0$, there exists $n_0 \in \mathbb{N}$ such that the following holds for all $n \geqslant n_0$. If G is an n-vertex graph with $\delta(G) \geqslant (1+\alpha)\frac{n}{2}$, then G contains a copy of every n-vertex tree with maximum degree bounded by Δ.*

Theorem 4.2 admits a spread version, as demonstrated in [154].

Theorem 4.3 (Pham, Sah, Sahwney, Simkin [154]). *For every $\Delta \in \mathbb{N}$ and $\alpha > 0$, there exists $n_0, C \in \mathbb{N}$ such that the following holds for all $n \geqslant n_0$. If G is an n-vertex graph with $\delta(G) \geqslant$*

$(1+\alpha)\frac{n}{2}$, *and* T *is a* n*-vertex tree with* $\Delta(T) \leqslant \Delta$*, there exists a* (C/n)*-spread distribution on embeddings of* T *onto* G*.*

Using the $s = n$ case of Definition 4.1, Theorem 4.3 allows us to deduce that in the context of Theorem 4.2, G contains $\Omega(n)^n$ copies of a given bounded degree tree (see [102] for a more precise result). Furthermore, Theorem 4.3 implies that the random subgraph $G' \subseteq G$ obtained by keeping each edge of G with probability $\Omega(\log n/n)$ also contains a given bounded degree tree (see [154] for a precise statement).

The original proof [121] of Theorem 4.2 constitutes one of the early applications of the Szemerédi regularity lemma (used in conjunction with the blow-up lemma of Komlós, Sárközy, and Szemerédi). The proof of the more general Theorem 4.3 in [154] can be interpreted as a randomised version of the proof in [121]. Indeed, readers familiar with applications of the regularity/blow-up lemma would know that whilst embedding a target subgraph with this method, there is actually a lot of flexibility for where each vertex can go. Thus, a choice can be made randomly from the available options as a reasonable strategy towards proving Theorem 4.3.

The main focus of the current chapter is a proof of Theorem 4.3 that uses Theorem 4.2 as a black-box. The most obvious advantage of such a proof is that, as Theorem 4.2 has a more modern proof due to Montgomery and Kathapurkar [111] that circumvents the use of Szemerédi regularity lemma, our proof yields a regularity-free proof of Theorem 4.3 which naturally has better dependencies between the constants (see Remark 4.14). Our proof is presented in Section 4.3, and Section 4.3.1 contains an overview explaining the key ideas.

Our methods are fairly general, and flexible. We believe they could translate to spread distribution in other related structures such as directed trees or hypertrees, and maybe to other structures admitting nice decomposition rules, such as spanning grids. we give more details about this intuition in Section 4.4.

4.2 Preliminaries

We use the standard notation for hierarchies of constants, writing $x \ll y$ to mean that there exists a non-decreasing function $f : (0,1] \to (0,1]$ such that the subsequent statements hold for $x \leqslant f(y)$. Hierarchies with multiple constants are defined similarly.

We will use the following theorem to embed our subtrees of bounded size. It generalises Theorem 4.2 and was proved in [145], using tools from [111]. In particular, the proof of the following theorem does not rely on the Szemerédi regularity lemma.

Theorem 4.4 (Theorem 4.4 [145])**.** *Let* $1/n \ll 1/\Delta, \alpha$*. Let* G *be an* n*-vertex graph with* $\delta(G) \geqslant (1/2+\alpha)n$*. Let* T *be an* n*-vertex tree with* $\Delta(T) \leqslant \Delta$*. Let* $t \in V(T)$ *and* $v \in V(G)$*. Then,* G *contains a copy of* T *with* t *copied to* v*.*

4.2.1 Tree-splittings

Definition 4.5. *Let T be an n-vertex tree. A* tree-splitting *of size ℓ is a family of edge-disjoint subtrees $(T_i)_{i\in[\ell]}$ of T such that $\bigcup_{i\in[\ell]} E(T_i) = E(T)$. Note that for any $i \neq j$, the subtrees T_i and T_j intersect on at most one vertex. Given a tree-splitting $(T_i)_{i\in[\ell]}$ of an n-vertex tree T, we define the* bag-graph *of the tree-splitting to be the graph whose nodes are indexed by $[\ell]$ and in which the nodes i and j are adjacent if $V(T_i) \cap V(T_j) \neq \emptyset$. A* bag-tree *of a tree-splitting is simply a spanning tree of the bag-graph.*

We will use the following simple proposition to divide a tree into subtrees (see, for example, [144, Proposition 3.22]).

Proposition 4.6. *Let $n, m \in \mathbb{N}$ such that $1 \leqslant m \leqslant n/3$. Given any n-vertex tree T containing a vertex $t \in V(T)$, there are two edge-disjoint trees $T_1, T_2 \subset T$ such that $E(T_1) \cup E(T_2) = E(T)$, $t \in V(T_1)$ and $m \leqslant |T_2| \leqslant 3m$.*

This implies that a tree can be divided into many pieces of roughly equal size, as follows (see [145] for the simple proof).

Corollary 4.7. *Let $n, m \in \mathbb{N}$ satisfy $m \leqslant n$. Given any n-vertex tree T, there exists a tree-splitting $(T_i)_{i\in[\ell]}$ of T such that for each $i \in [\ell]$, we have $m \leqslant |T_i| \leqslant 4m$.*

4.2.2 Probabilistic results

Below we give a lemma that encapsulates a simple argument often used when computing the spreadness of a random permutation.

Lemma 4.8. *Let $n \in \mathbb{N}$, $s \leqslant n$, and $L_1, \dots, L_s \subseteq [n]$. For any distinct integers $1 \leqslant x_1, \dots, x_s \leqslant n$, a uniformly sampled permutation π of $[n]$ satisfies $\mathbb{P}\left[\bigwedge_{i=1}^{s} \pi(x_i) \in L_i\right] \leqslant \prod_{i=1}^{s} \frac{e|L_i|}{n}$.*

Proof of Lemma 4.8. We have the following,

$$\mathbb{P}\left[\bigwedge_{i=1}^{s} \pi(x_i) \in L_i\right] = \prod_{i=1}^{s} \mathbb{P}\left[\pi(x_i) \in L_i \,\middle|\, \bigwedge_{j=1}^{i-1} \pi(x_j) \in L_j\right] \leqslant \prod_{i=1}^{s} \frac{|L_i|}{n-i+1} \leqslant \prod_{i=1}^{s} \frac{e|L_i|}{n},$$

where in the last step we used the fact that $\prod_{i=1}^{s} n - i + 1 \geqslant \left(\frac{n}{e}\right)^s$, which is a well-known application of Stirling's approximation. □

Next, we present a lemma that records several properties we need from a random vertex partition of a dense graph. The proof consists of standard applications of well-known concentration inequalities, namely Chernoff's bound and McDiarmid's inequality, and can be found in the complete version of this result [30].

Lemma 4.9. *Let $1/n \ll \eta, 1/C, 1/K, \delta, \alpha$, and suppose $1/C \ll 1/K \ll \alpha$. Fix two sequences of integers $(a_c)_{C \leqslant c \leqslant 4C} \geqslant 1$ and $(b_c)_{C \leqslant c \leqslant 4C} \geqslant 1$ such that $\sum_{C \leqslant c \leqslant 4C} b_c a_c < n$, $b_c \geqslant \eta n$ and $C - K \leqslant a_c \leqslant 4C - K$ for each $C \leqslant c \leqslant 4C$. Set $\varepsilon := e^{-\alpha^2 C/12}$. Let G be a n-vertex graph with $\delta(G) \geqslant (\delta + \alpha)n$ and let $v \in V(G)$. Then, there exists a random labelled partition $\mathcal{R} = (R_c^j)_{C \leqslant c \leqslant 4C, j \in [b_c]}$ of a subset of $V(G) \setminus \{v\}$ into $\sum_{C \leqslant c \leqslant 4C} b_c$ parts, with the following properties,*

A1 *$\forall c \in [C, 4C], \forall j \in [b_c], |R_c^j| = a_c$, meaning there are exactly b_c parts of size a_c;*

A2 *$\forall R_c^j \in \mathcal{R}$, $|\{u \in V(G) \mid \deg(u, R_c^j) \geqslant (\delta + \frac{\alpha}{2})|R_c^j + u|\}| \geqslant (1 - 3e^{-\frac{\alpha^2 C}{10}})|V(G)|$;*

A3 *$\forall u \in V(G)$, $|\{R_c^j \in \mathcal{R} \mid \delta(G[R_c^j + u]) \geqslant (\delta + \frac{\alpha}{2})|R_c^j|\}| \geqslant (1 - 3e^{-\frac{\alpha^2 C}{10}})|\mathcal{R}|$.*

Moreover, call $R_c^j \in \mathcal{R}$ good if $\delta(G[R_c^j]) \geqslant (\delta + \alpha/2)|R_c^j|$. Call $R_c^j, R_d^k \in \mathcal{R}$ a good pair if $\delta(G[R_c^j + v]) \geqslant (\delta + \alpha/2)|R_c^j + v|$ for each $v \in R_d^k$, and the same statement holds with j and c interchanged with k and d. Let A be the auxiliary graph with vertex set $\mathcal{R}$ where $R_c^j \sim_A R_d^k$ if and only if R_c^j, R_d^k is a good pair. Then, there exists a subgraph A' of A such that

B1 *For all $c \in [C; 4C]$, $A'_c := \{R_c^j \in A' : |R_c^j| = a_c\}$ has size at least $(1 - \varepsilon)b_c$;*

B2 *$\forall R_c^j \in A', \forall d \in [C, 4C], \deg_{A'}(R_c^j, A'_d) \geqslant (1 - \varepsilon)|A'_d|$;*

B3 *$\forall R_c^j \in A'$, R_c^j is good.*

Furthermore, the following spreadness property holds. For any function $f \colon \{u_1, \dots, u_s\} \to \mathcal{R}$ (where $\{u_1, \dots, u_s\} \subseteq V(G)$), $\mathbb{P}[u_i \in f(u_i)] \leqslant (12C/n)^s$.

4.2.3 Good spread with high minimum degree

If the minimum degree of the host graph is larger than the size of the target graph we wish to embed, a simple random greedy algorithm can find a distribution with good spread. The following lemma records a version of this where vertices of the target and host graphs are coloured, and the embedding we produce respects the colour classes. This is used in the proof whilst (randomly) embedding the bag tree of a tree splitting into an auxiliary graph where vertices represent random subsets of a host graph, edges represent good pairs (as in Lemma 4.9), and the colours represent the size of the random set.

Lemma 4.10. *Let $1/n \ll \gamma \ll \eta \leqslant 1$. Let G be a graph and $v \in V(G)$. Let $V_1 \cup \ldots \cup V_k$ be a partition of $V(G) \setminus \{v\}$ where $|V_i| \geqslant \eta n$ for all $i \in [k]$ and so that for all $i \in [k]$, for all $u \in V(G)$, $\deg(u, V_i) \geqslant (1 - \gamma)|V_i|$. Let T be a tree and let t be a vertex of T. Let c be a k-colouring of $T - t$ such that the number of vertices coloured i is at most $(1 - \eta)|V_i|$. Then, there exists a random embedding $\phi \colon T \to G$ such that the following all hold.*

1. *With probability* 1, $\phi(t) = v$.
2. *With probability* 1, *any* i*-coloured vertex of* T *is embedded in* V_i.
3. *The random embedding induced via* ϕ *by restricting to the forest* $T - t$ *is a* $\left(\frac{2}{\eta^2 n}\right)$*-spread embedding.*

Proof. Let us consider an ordering $t, t_1, t_2, \ldots, t_m$ of the vertices of T rooted in t, where $T[\{t, t_1, \ldots, t_i\}]$ is a subtree for each $i \in [m]$. We define $\phi : T \to G$ greedily vertex by vertex following the ordering of $V(T)$. Let $\phi(t) := v$. We denote by p_i the parent of t_i in T for all $i \geqslant 1$, and define $\phi(t_i)$ conditioned on some value of $\phi(t), \phi(t_1), \ldots \phi(t_{i-1})$ to be the random variable following the uniform distribution over $\left(V_{c(t_i)} \cap N(\phi(p_i))\right) \setminus \cup_{j=1}^{i-1} \phi(t_j)$. Note that,

$$\begin{aligned}\left|\left(V_{c(t_i)} \cap N(\phi(p_i)\right) \setminus \cup_{j=1}^{i-1}\{\phi(t_j)\}\right| &\geqslant \deg(p_i, V_{c(t_i)}) - |c^{-1}(c(t_i))| \\ &\geqslant ((1-\gamma) - (1-\eta))\,|V_{c(t_i)}| \\ &\geqslant (\eta - \gamma)\eta n.\end{aligned}$$

We now discuss the spreadness of ϕ. Fix an integer $s \leqslant k$ and two sequences $t'_1, \ldots, t'_s \in V(T) \setminus \{t\}$ and $v_1, \ldots, v_s \in V(G) \setminus \{v\}$ of distinct elements. Moreover, let us suppose that $t'_1, t'_2, \ldots, t'_s$ appear in this order in the ordering chosen above. Observe that

$$\mathbb{P}\left[\bigwedge_{i=1}^{s} \phi(t'_i) = v_i\right] = \prod_{i=1}^{s} \mathbb{P}\left[\phi(t'_i) = v_i \,\middle|\, \bigwedge_{j=1}^{i-1} \phi(t'_j) = v_j\right] \leqslant \left(\frac{1}{(\eta-\gamma)\eta n}\right)^s.$$

The last inequality follows as for any $u \in V(G)$ and $i \in [k]$ the probability that $\phi(t_i) = u$ conditioned on any value of $\phi(t), \phi(t_1), \ldots, \phi(t_{i-1})$ is at most $\frac{1}{(\eta-\gamma)\eta n}$. In particular, the lemma follows as we have $\gamma \leqslant \eta/2$. □

4.3 Proof of Theorem 4.3

4.3.1 Overview

As briefly discussed in Section 4.4, our proof capitalises on several desirable properties (as collected in Lemma 4.9) satisfied by a random partition of the vertex-set of the host graph G. In this way, our proof bears resemblance to the proof in [115]; however, we emphasise that the specific constructions of the distributions are otherwise quite different. In particular, the key idea in the current work could be used to give a distinct, and more concise proof of the results in [115].

To start, we split T into $O(1)$-sized edge-disjoint subtrees via Corollary 4.7 and take a random partition $\mathcal{R}$ of the host graph given by Lemma 4.9 (we will comment on the choice of parameters momentarily). Almost all of the random subsets R in $\mathcal{R}$ have good enough minimum degree to contain all bounded degree trees of size $|R|$ by just applying Theorem 4.2 as a black-box. Now, we need to decide (randomly), which subtrees of T will embed into which random subsets of $V(G)$. This corresponds to randomly embedding a bag-tree of the tree-splitting into A', the auxiliary graph given in Lemma 4.9 that encodes the pairs of random sets with good minimum degree. Thus, we reduce Theorem 4.3 to a weaker version of itself where the host graph G is nearly complete (thanks to **B2**). Unfortunately, we do not have a way of directly producing the necessary random embedding even in this simpler context where the minimum degree of the host graph is extremely large[1].

To circumvent this problem, we introduce the following trick which we hope might have further applications. While applying Lemma 4.9, we make the sizes of the random sets an ε-fraction smaller than the sizes of the subtrees they are meant to contain. This gives us extra space as we then have more random sets than subtrees we need to embed. Afterwards, using a simple random greedy strategy (see Lemma 4.10), we can produce the necessary random embedding ψ of the bag-tree into A'. Two problems remain: some random sets are unused by ψ and the random sets that are used by ψ are too small to contain the subtrees we wish to embed. We fix both of these issues by randomly reallocating all vertices of the unused random sets into the used random sets using Corollary 4.12. We need a fair bit of precision in this final step, which is discussed more in Section 4.3.2.

To finish the embedding, we need to convert ψ into a random embedding $\phi : V(T) \mapsto V(G)$. We may do this by ordering the subtrees so that each subtree intersects the previous ones in at most one vertex, and using Theorem 4.4, which is a slight strengthening of Theorem 4.2 that allows us to prescribe the location of a root vertex in advance. To illustrate, suppose $T_1, \dots, T_{i-1}$ are already embedded, and suppose that $\psi(T_i)$ is an empty random set large enough to contain T_i. Say there exists some $t \in V(T_i) \cap V(T_j)$ for some $j < i$, then $\phi(t)$ is already determined, as T_j is already embedded. The properties of ψ, coming from Lemma 4.9 and Lemma 4.10, guarantee that $\phi(t) \cup \psi(T_i)$ has large minimum degree, so we may invoke Theorem 4.4 to extend ϕ to embed vertices of T_i in $\psi(T_i)$, respecting the previous choice of $\phi(t)$, as desired.

[1] In contrast, a similar method is employed in [115] to embed hypergraph Hamilton cycles, but here the "bag-tree" of a hypergraph Hamilton cycle is simply a 2-uniform Hamilton cycle, which is simpler to embed randomly with good spread using elementary methods.

4.3.2 Spread distributions on star matchings

As described in Section 4.3.1, we need a way to randomly shuffle around vertices in a random partition of G to adjust the sizes of certain random sets, and we need to do this without damaging the randomness properties or the minimum degree conditions of the partition (coming from Lemma 4.9). The following lemma gives us a way to achieve this in the special case where each random set is meant to receive exactly one new element.

Lemma 4.11 (Lemma 3.1 in [115]). *There exists an absolute constant $C_{4.11}$ with the following property. If G is a balanced bipartite graph on $2n$ vertices with $\delta(G) \geqslant 3n/4$, then there exists a random perfect matching M of G such that for any collection of edges $e_1, \dots, e_s \in E(G)$, $\mathbb{P}[\bigwedge_{i\in[s]} e_i \in M] \leqslant (C_{4.11}/n)^s$.*

As we make each original random set multiple vertices smaller than it needs to be to be able to contain the corresponding subtree (recall Section 4.3.1), each random set actually has to receive more than just a single new element. The next result generalises the previous lemma into this context.

Corollary 4.12. *There exists an absolute constant $C_{4.12}$ with the following property. Let $1/n \ll 1/k$. Let G be a bipartite graph with partition (A, B), with $|A| = n, |B| = kn$. Suppose for each $a \in A$, $d(a, B) \geqslant (99/100)|B|$ and for each $b \in B$, $d(b, A) \geqslant (99/100)|A|$. Then, there is a random $K_{1,k}$-perfect-matching M of G (where the centres of the $K_{1,k}$ are embedded in A) such that for any collection of edges $e_1, \dots, e_s \in E(G)$, $\mathbb{P}[\bigwedge_{i\in[s]} e_i \in M] \leqslant (C_{4.12}/n)^s$.*

Proof. There exists an equipartition of B as $B_1, \dots, B_k$ such that each $G[A \cup B_i]$ ($i \in [k]$) is a graph with minimum degree at least $(98/100)n$. Indeed, a random partition of B would have this property with high probability (as $n/k \to \infty$, see, for example, Lemma 3.5 from [87]). Now, Lemma 4.11 gives us a random perfect matching M_i in each $G[A \cup B_i]$ ($i \in [k]$), and $\bigcup_{i\in[k]} M_i =: M$ is a random $K_{1,k}$-perfect-matching of G. M clearly has the desired spread with $C_{4.12} = C_{4.11}$. □

4.3.3 Proof of Theorem 4.3

We actually prove a stronger version of Theorem 4.3 where the location of a root vertex is specified in advance (similar to Theorem 4.4) as we believe this stronger result could have further applications. The unrooted version, i.e. Theorem 4.3, follows simply by choosing $t \in V(T)$ arbitrarily, and $v \in V(G)$ uniformly at random, setting $\phi(t) = v$, and using Theorem 4.13 to complete this to a full $O(1/n)$-spread embedding of T.

Theorem 4.13. *Let $\frac{1}{n} \ll \frac{1}{C_*} \ll \alpha, 1/\Delta$. Let G be a n-vertex graph with $\delta(G) \geqslant (1/2 + \alpha)n$. Let T be a tree on n vertices, with $\Delta(T) \leqslant \Delta$. Let $t \in V(T)$ and let $v \in V(G)$. Then, there*

exists a random embedding $\phi\colon T \to G$ such that $\phi(t) = v$ with probability 1 and ϕ restricted to $T - t$ is $\left(\frac{C_*}{n}\right)$-spread.

Remark 4.14. All of the dependencies between the constants that arise from our proof are polynomial. However, we also need that C_* is at least a polynomial function in $f(\alpha, \Delta)$, where f is the function from Theorem 4.4 that ultimately relies on [111]. Unfortunately, [111] does not cite an explicit bound (though their proof does not use the Szemerédi regularity lemma).

Proof of Theorem 4.13. Let C be a new constant such that $1/n \ll 1/C_* \ll 1/C \ll \alpha, 1/\Delta$. C
Let $(T_i)_{i\in[\ell]}$ be a tree-splitting of T obtained by Corollary 4.7 applied with $m := C$. Notice that adding $T_* := \{t\}$ to this tree-splitting produces another tree-splitting. Let T' be a bag-tree of $(T_i)_{i\in[\ell]} \cup \{T_*\}$ rooted in T_*. Note that T' has maximum degree $4C\Delta$. T_* T'

Step 1: Randomly partition of $V(G) \setminus \{v\}$. We assign to each subtree T_i a colour that corresponds to its size. Formally, define the colouring $f : V(T')\setminus\{T_*\} \to [C, 4C]$ via $f(T_i) := |T_i|$ for each subtree T_i. Fix an integer K such that $1/C \ll 1/K \ll \alpha$. For each colour $c \in [C, 4C]$, let $a_c := c - 1 - K$ and $b_c := |f^{-1}(c)| + \left\lfloor \frac{K}{32C^3} n \right\rfloor$. f K, a_c, b_c

We use Lemma 4.9 on G with the following parameters, $(a_i)_{i\in[k]} := (a_c)_{C\leqslant c\leqslant 4C}$, $(b_i)_{i\in[k]} := (b_c)_{C\leqslant c\leqslant 4C}$, $\delta := \frac{1}{2}$, $K := K+1$, and $C, \alpha, v := C, \alpha, v$. To do so, we need only need to check that $\sum_c a_c b_c < n$, as the other conditions follow directly from our choice of constants. Observe

$$\sum_{c=C}^{4C} a_c b_c = \sum_{i=1}^{\ell} (|T_i| - 1 - K) + \left\lfloor \frac{K}{32C^3} n \right\rfloor \sum_{c=C}^{4C} (c - 1 - K) < |T| - \ell K + \left\lfloor \frac{K}{32C^3} n \right\rfloor \frac{5C(3C+1)}{2} \leqslant n$$

We can thus obtain a partition $\mathcal{R} = (R_c^j)_{C\leqslant c\leqslant 4C, j\in[b_c]}$ of a subset of $V(G) \setminus \{v\}$ and an auxiliary graph A' whose vertex set $V(A')$ is a subset of $\mathcal{R}$, satisfying the conditions listed in Lemma 4.9. Add to A' a vertex $R_* = \{v\}$ adjacent to all $R_c^j \in \mathcal{R}$ such that $\delta(G[R_c^j + v]) \geqslant \left(\frac{1}{2} + \frac{\alpha}{2}\right) |R_c^j + v|$. For each colour $c \in [C, 4C]$, let us denote by $V_c := \{R_c^j \mid j \in [b_c]\} \cap V(A')$. We define the colouring $g : V(A') \mapsto [C, 4C]$ that associates the colour c to all parts in V_c. Formally, $\forall c \in [C, 4C], \forall R_c^j \in V_c, g(R_c^j) = c$. R_* V_c, g

Step 2: Construct $\psi\colon T' \to A'$. Conditional on a fixed outcome of $\mathcal{R}$ (and thus, A'), we describe ψ, which is a random embedding of T' on A'. We apply Lemma 4.10 to A' with partition $V_C \cup \ldots \cup V_{4C}$ and T' coloured by f, with parameters $t := T_*$, $v := R_*$, $\gamma := e^{-\alpha^2 C/12}$ and $\eta := \left\lfloor \frac{K}{32C^3} \right\rfloor$ to obtain a random embedding ψ. To apply the lemma, we need the following conditions to be satisfied: ψ γ, η

- $1/n \ll \gamma = e^{-\alpha^2 C/12} \ll \eta = \Theta\left(\frac{K}{C^3}\right)$,
- for all $i \in [C; 4C]$, for all $u \in V(A')$, $\deg(u, V_i) \geqslant (1-\gamma)|V_i|$,
- for all c, the number of subtrees coloured c is at most $(1-\eta)|V_c|$.

The first condition is satisfied by our constant hierarchy. Our choice of γ and condition **B1** of Lemma 4.9 is tailored so that $V(A')$ satisfies the second condition. The third condition is less direct. Let n_c be the number of subtrees coloured c in T', what we aim to show is $n_c \leqslant (1-\eta)|V_c|$. By Condition **B1** of Lemma 4.9 and the definition of b_c, we have that $|V_c| \geqslant (1 - e^{-\frac{\alpha^2 C}{12}})(n_c + \eta n)$. Hence,

$$\begin{aligned}(1-\eta)|V_c| &\geqslant (1-\eta)(1-e^{-\frac{\alpha^2 C}{12}})(n_c+\eta n) \geqslant (1-\eta-e^{-\frac{\alpha^2 C}{12}})(n_c+\eta n) \geqslant (1-2\eta)(1+\frac{\eta n}{n_c})n_c \\ &\geqslant (1-2\eta)(1+C\eta)n_c \geqslant n_c\end{aligned}$$

where we used that $e^{-\frac{\alpha^2 C}{12}} \ll \eta = \Theta\left(\frac{K}{C^3}\right)$, and for the last step that $n_c \leqslant \frac{n}{C}$ and $C \geqslant 4$.

Recall that $\psi : V(T') \mapsto V(A')$ denotes the random embedding obtained from Lemma 4.10. By construction, ψ restricted to $T' \setminus \{T_*\}$ is $(\frac{2^{15}C^6}{n})$-spread, and note that this spreadness condition holds independently of the values of $\mathcal{R}$ and A' that we condition on. Lemma 4.10 also ensures that with probability 1, ψ preserves the colouring given by g, i.e. $\forall i \in [\ell], f(T_i) = g(\psi(T_i))$.

Step 3: Adjust the size of the bags. In this step, we describe how to obtain a randomised partition $\mathcal{M}$ of $V(G)$. We define $\mathcal{M}$ conditional on fixed values of $\mathcal{R}$ and ψ. Informally, our goal is to build, for all $R_c^j \in \mathrm{Im}(\psi)$, a set M_c^j satisfying $R_c^j \subseteq M_c^j$ and $|M_c^j| = c-1$ while preserving the minimum degree condition given by Lemma 4.9 for the set $\mathrm{Im}(\psi)$ and the edges induced by ψ. Formally, defining $N(R_c^j) = \{R \in \mathrm{Im}(\psi) | \{\psi^{-1}(R_c^j), \psi^{-1}(R)\} \in E(T')\}$, we want $\mathcal{M}$ to satisfy the following three properties:

$N(R_c^j)$

C1 $\forall R_c^j \in \mathrm{Im}(\psi), |M_c^j| = c-1$;

C2 $\forall R_c^j \in \mathrm{Im}(\psi), \delta(G[M_c^j]) \geqslant \left(\frac{1}{2} + \frac{\alpha}{3}\right)|M_c^j|$;

C3 $\forall R_c^j \in \mathrm{Im}(\psi), \forall R_{c'}^{j'} \in N(R_c^j), \forall v \in M_c^j, \delta\left(G\left[M_{c'}^{j'} + v\right]\right) \geqslant \left(\frac{1}{2} + \frac{\alpha}{3}\right)|M_{c'}^{j'} + v|$.

H, A, B

Consider the following bipartite graph H with bipartition (A, B) where $B = V(G) \setminus (\{t\} \cup \cup_{R \in \mathcal{R}} R)$ and $A = \mathcal{R}$. Put an edge between $v \in A$ and $R \in B$ if $\forall R' \in \{R\} \cup N(R)$, $\delta(G[R' + v]) \geqslant \left(\frac{1}{2} + \frac{\alpha}{2}\right)|R' + v|$. Note that, for all $(a, b) \in A \times B$, **A2** and **A3** imply that

$d(a,B) \geqslant (1-(4\Delta C+1)3e^{-\frac{\alpha^2 C}{10}})|B| \geqslant \frac{99}{100}|B|$ and that $d(b,A) \geqslant (1-(4\Delta C+1)3e^{-\frac{\alpha^2 C}{10}}) \geqslant \frac{99}{100}|B|$. Therefore, we may use Corollary 4.12 on H with $k := K$, to associate to each R_c^j a disjoint random set of K elements of B, satisfying the spreadness property stated in the lemma (regardless of the value of $\mathcal{R}$ and ψ that is being conditioned upon). Consider $L_c^j \subseteq A$, the set of random vertices that are matched by the $K_{1,K}$-perfect-matching to R_c^j, and define $M_c^j := R_c^j \cup L_c^j$. Note **C1** is then directly satisfied. The definition of an edge in H and the fact that $K/C \ll \alpha$ imply that **C2** and **C3** are also both satisfied. We define $\mathcal{M} := \bigcup_{c,j}\{M_c^j\}$. L_c^j M_c^j $\mathcal{M}$

Having defined the random variable $\mathcal{M}$, we now show the following spreadness property. To clarify, ψ and $\mathcal{R}$ are not considered to be fixed anymore.

Claim 4.15. *For any $s \in \mathbb{N}$ and any function $h : \{v_1,\dots,v_s\} \mapsto \mathcal{M}$ where $\{v_1,\dots,v_s\} \subseteq V(G)$, we have $\mathbb{P}\left[\wedge_{i=1}^s v_i \in h(v_i)\right] \leqslant (\frac{12C\cdot C_{4.12}}{n})^s$.*

Proof. Note that if $v \in \{v_1,\dots,v_s\}$ then $\mathbb{P}[v \in h(v)] = 0$. Suppose this is not the case, and let us partition $\{v_1,\dots,v_s\}$ into $\{x_1,\dots,x_{s_1}\} \subseteq A$ and $\{y_1,\dots,y_{s_2}\} \subseteq V(G)\setminus(A\cup\{v\})$. Observe that

$$\mathbb{P}\left[\bigwedge_{i=1}^{s} v_i \in h(v_i)\right] = \mathbb{P}\left[\bigwedge_{i=1}^{s_2} y_i \in h(y_i)\right]\mathbb{P}\left[\bigwedge_{i=1}^{s_1} x_i \in h(x_i) \,\middle|\, \bigwedge_{i=1}^{s_2} y_i \in h(y_i)\right] \leqslant \left(\frac{12C}{n}\right)^{s_2}\left(\frac{C_{4.12}}{n}\right)^{s_1},$$

where we used the spreadness property from Lemma 4.9 and Corollary 4.12 in the last step. □

Step 4: Embed the subtrees. From now on, we redefine g and ψ as being maps to $\mathcal{M}\cup\{R_*\}$ (by composing g and ψ with the natural bijection $\mathcal{R} \mapsto \mathcal{M}\cup\{R_*\}$ that associates M_c^j to R_c^j, and R_* to itself).

We fix $\phi(t) := v$, by doing so we embed T_* into R_*. The goal is now to embed each T_i in $\psi(T_i) \in \mathcal{M}$. Note that $|\psi(T_i)| = |T_i| - 1$ for all $i \in [\ell]$ (by **C1**). We define ϕ as follows: While there exists a subtree T_i that is not fully embedded into G, pick a subtree T_i that has exactly one vertex t_i already embedded say $\phi(t_i) = v_i$ and apply Theorem 4.4 to embed the rest of T_i in $\psi(T_i)$. We can use Theorem 4.4, due to **C1**, **C2** and **C3**. This procedure is well defined because T' is a tree. Let us define the *native atom* of a vertex $y \in V(T)$, denoted by $T(y)$, to be the first T_i that contains y. $T(y)$

Checking spreadness. We now prove that the random embedding ϕ constructed this way is $\left(\frac{C_*}{n}\right)$-spread. The spreadness of this embedding comes from two different randomness sources: the partition $\mathcal{M}$ via Claim 4.15, and the random embedding ψ via Lemma 4.10.

Let us fix two sequences of distinct elements $y_1,\dots,y_s \in V(G-v)$ and $x_1,\dots x_s \in V(T-t)$. Let $b := |\{T(x_i) \mid i \in [s]\}|$. We may suppose, up to reordering, that $x_1,\dots,x_b$ each have distinct native atoms, this way we have $\{T_{x_1},\dots,T_{x_b}\} = \{T_{x_1},\dots,T_{x_s}\}$. Let us split our probability on the two sources of spreadness as follows. Set $C_0 := 12C\cdot C_{4.12}$.

$$\begin{aligned}
\mathbb{P}\left[\bigwedge_{i=1}^{s}\phi(x_i)=y_i\right] &= \sum_{h:[b]\mapsto\mathcal{M}}\mathbb{P}\left[\bigwedge_{i=1}^{s}\phi(x_i)=y_i\middle|\bigwedge_{i=1}^{s}y_i\in h(i)\right]\cdot\mathbb{P}\left[\bigwedge_{i=1}^{s}y_i\in h(i)\right]\\
&\leqslant \sum_{h:[b]\mapsto\mathcal{M}}\mathbb{P}\left[\bigwedge_{i=1}^{s}\phi(x_i)=y_i\middle|\bigwedge_{i=1}^{s}y_i\in h(i)\right]\cdot\left(\frac{C_0}{n}\right)^s && \text{by Claim 4.15}\\
&\leqslant \sum_{h:[b]\mapsto\mathcal{M}}\mathbb{P}\left[\bigwedge_{i=1}^{b}\psi(T(x_i))=h(i)\right]\cdot\left(\frac{C_0}{n}\right)^s\\
&\leqslant \sum_{h:[b]\mapsto\mathcal{M}}\left(\frac{2C^2}{\eta^2 n}\right)^b\left(\frac{C_0}{n}\right)^s && \text{by Lemma 4.9}\\
&\leqslant |\mathcal{M}|^b\left(\frac{2C^2}{\eta^2 n}\right)^b\left(\frac{C_0}{n}\right)^s\\
&\leqslant n^b\left(\frac{2C^2}{\eta^2 n}\right)^b\left(\frac{C_0}{n}\right)^s\\
&\leqslant \left(\frac{2C^2C_0}{\eta^2 n}\right)^s\leqslant\left(\frac{C_*}{n}\right)^s && b\leqslant s
\end{aligned}$$

To justify the second inequality, it is sufficient to observe that $\phi(x_i)=y_i$ only if $\psi(T(x_i))=h(i)$. Moreover by the remark made above, $T(x_1),\dots,T(x_b)$ are all distinct, so we can indeed invoke the spreadness property of ψ coming from Lemma 4.10. □

4.4 Conclusion and perspectives

Our proof could be interpreted as modest progress towards a more ambitious research agenda, hinted to in [115], which asserts that *all* Dirac-type results admit a spread version, regardless of the target structure being embedded. One reason why such a general result could hold is that in Dirac-type results, host graphs have linear minimum degree. Thus, Chernoff's bound can be used to show that almost all $O(1)$-sized induced subgraphs of such dense host graphs maintain the same (relative) minimum degree condition. If the target graph itself has some recursive structure, we may use this to our advantage whilst constructing a random embedding with optimal spread. The strategy would be to first break up the target graph into pieces of size $O(1)$, for example, in the case of Hamilton cycles, we would simply break up the cycle into several subpaths. For each $O(1)$-sized subpath, almost all $O(1)$-sized subsets of the host graph have large enough minimum degree to necessarily

contain a copy of the subpath (simply by invoking Dirac's theorem), so we may choose one such host subset randomly while constructing a random embedding with good spread.

A variant of the above strategy was successfully implemented in [115] in the context of hypergraph Hamilton cycles. In this chapter, we devise a novel strategy that works for bounded-degree spanning trees, which, like Hamilton cycles, have a recursive structure, albeit a lot more complex than that of Hamilton cycles. In particular, a Hamilton cycle can be thought of as a union of subpaths of essentially the same length, whereas any partition of a tree into further subtrees needs to use subtrees of a wide range of possible sizes (see Section 4.2.1), which makes it difficult to extend the techniques of [115] to bounded-degree trees.

Our proof is fairly short; however, we explain the key idea in Section 4.3.1. We believe our methods are fairly general, and they could translate to construct spread distributions in the context of directed trees [111], hypertrees [152, 151], or other related structures such as spanning grids.

It remains an interesting open problem to find an even larger class of target graphs for which the Dirac-type theorem admits a spread generalisation. For example, it would be natural to investigate graph families with sublinear bandwidth, and we believe our methods could be applicable here. Note that this would entail more than simply randomising the blow-up lemma based proof of the Bandwidth Theorem [40], as this theorem does not always give the optimal minimum degree condition for the containment of every graph family with sublinear bandwidth. Though, of course, obtaining a spread version of the bandwidth theorem would be of independent interest.

Komlós, Sárközy, Szemerédi [122] actually proved a stronger result than Theorem 4.2 where the maximum degree hypothesis is relaxed as $\Delta(T) = o(n/\log n)$. It would be interesting to similarly strengthen Theorem 4.3 by weakening the assumption on $\Delta(T)$.

Chapter 5

Faithful universal structures

This chapter focuses on universal structures. A structure is called universal for a family $\mathcal{F}$ if it contains all the elements of $\mathcal{F}$ as a substructure. We study the case of universal posets and graphs. This chapter is based on two unpublished works: one with Carla Groenland and Rajko Nenadov, and the other with Carla Groenland, Louis Esperet, Claire Hilaire, Clément Rambaud, and Alexandra Wesolek.

5.1 Introduction

A family of graphs $\mathcal{U} = (U_n)_{n\in\mathbb{N}}$ is said to be *strongly-universal* for a graph class $\mathcal{G}$ if for any $n \in \mathbb{N}$, U_n contains all the n-vertex graphs of $\mathcal{G}$ as induced subgraphs. Similarly, $\mathcal{U}$ is said to be *weakly-universal* for $\mathcal{G}$ if for any $n \in \mathbb{N}$, U_n contains all the n-vertex graphs of $\mathcal{G}$ as subgraphs.

The initial motivation for the introduction of these concepts was the design of configurable chips [36, 57], with the idea that a single chip could be used to produce a number of different chips, by simply removing connectors (edges) or components (vertices) in a post-processing phase. This led to a considerable amount of work, trying to either minimise the number of edges in weakly-universal graphs [53, 55, 35, 17, 180, 34, 45, 10, 9, 70], or minimise the number of vertices in induced universal graphs [54, 80, 39, 67, 109, 8, 171, 147, 12, 13, 1, 10, 70].

For strongly-universal graphs, we give a sample of the best known results below. By a slight abuse of notation, when we say that for some function f, *a graph class $\mathcal{G}$ has weakly-universal (or strongly-universal) graphs of size at most $f(n)$*, we mean that $\mathcal{G}$ has a weakly-universal (or strongly-universal) graph family $\mathcal{U} = (U_n)_n \in \mathbb{N}$ such that each U_n contains

at most $f(n)$ vertices. With this terminology, there are strongly-universal graphs of order

- $O(n)$ for trees [12],
- $n^{1+o(1)}$ for graphs of bounded treewidth [80],
- $n^{1+o(1)}$ for planar graphs and a number of related graph classes [67, 81, 70],
- $n^{2+o(1)}$ for proper minor-closed classes [80],
- $O(n^{\Delta/2})$ for the class of graphs of maximum degree Δ [11], and
- $(1+o(1))2^{(n-1)/2}$ for the class of all graphs [8].

A classical problem related to strongly-universal graphs was the *Implicit graph conjecture* (see [109, 171]), which stated that every hereditary class of graphs which contains at most $2^{O(n \log n)}$ different n-vertex graphs admit a strongly-universal graph of polynomial size. This conjecture was only refuted recently by Hatami and Hatami [89].

There are two main techniques to construct universal graphs. In the case of strongly-universal graphs, researchers generally use the notion of an *adjacency labeling scheme*, introduced by Kannan, Naor and Rudich [109] and Muller [147]. Informally, this is a compact data structure describing the adjacency relation between vertices in a graph class. One weakness of such technique is that during the translation between the data structure and the universal graph all the structure of the original graphs is lost, therefore controlling the structure of the universal graphs is extremely difficult and even controlling the number of edges is difficult (see for instance [70]). In the case of weakly-universal graphs, best known bound usually come from separator theorems [53]. If the vertex set of any n-vertex graph $G \in \mathcal{G}$ can be partitioned into three sets X_1, S, X_2, with $|X_1| \leqslant 2n/3$, $|X_2| \leqslant 2n/3$, $|S| = O(n^{1-\varepsilon})$, and no edges between X_1 and X_2, then the idea is to take (inductively) two weakly-universal graphs for the graphs of size $|X_1|$ and $|X_2|$ in the class, and add a set of $|S|$ universal vertices. This typically creates large cliques, even if the graphs from $\mathcal{G}$ have a small clique number.

Let us say that a universal family $\mathcal{U} = (U_n)_{n\in\mathbb{N}}$ for a graph class $\mathcal{G}$ is *faithful* if for any $n \in \mathbb{N}$, $U_n \in \mathcal{G}$. In the context of chip design, the configurable chip has the same physical constraints as the different chips it is supposed to emulate. Thus in graph-theoretic terms we expect the underlying (universal) graphs to be faithful (for instance, if all graphs underlying the different chips have to be planar, the configurable chip also has to be planar). Note that as explained above, while minimizing the number of vertices in a weakly-universal in general is not relevant (because complete graphs contain all graphs of the same order as subgraphs), minimizing the number of vertices in a *faithful* weakly-universal graph is interesting, especially when the original graph class $\mathcal{G}$ is sparse (and thus does not contain large cliques).

Countable universal graphs Beyond the motivation of designing configurable chips, the concept of *faithful* countable graphs is well established (though under a different name) in mathematics and has received a lot of attention over the past 40 years [3, 120, 59, 119, 78, 79, 118, 49, 50, 47, 51, 48, 93]. In this context, we say that an (infinite) countable graph U is faithful strongly-universal for a class of countable graphs $\mathcal{G}$ if $U \in \mathcal{G}$ and U contains every graph of $\mathcal{G}$ as an induced subgraph. A well-known example of this notion is the *Rado graph*. Ackermann [3], Erdős and Rényi [69], and Rado [155] independently proved that the Rado graph is faithful strongly-universal for the class of all countable graphs. The Rado graph has, since then, proven to be useful in solving multiple other questions in combinatorics (see the following surveys [43, 44]). Henson [91] has proven the existence of a faithful strongly-universal graph for the class of countable K_t-free graphs. Since then, there has been a systematic study of faithful universal graph for the family of countable H-free graphs [120, 119, 78, 79, 118, 49, 50, 47, 51, 48]. Diestel, Halin and Vogler [60] proved that for any $t \leqslant 4$, there does not exist a faithful universal graphs for the family of countable K_t-minor-free graphs, and similar results are known when excluding $K_{s,t}$ has a minor [59].

In the context of minor-closed classes of graphs, Ulam asked the following natural question:

Problem 5.1. *Does there exist a faithful strongly-universal graph for the class of countable planar graphs?*

This question was answered negatively by Pach [148]. In his paper, Pach also asked whether the faithful condition could be relaxed to obtain a strongly-universal graph for the class of countable planar graphs while still preserving key properties of planar graphs. Following the direction of Pach's question, it is natural to ask if there exists a strongly-universal graph U for countable planar graphs, such that U is H-minor-free for some graph H. Forty years after Pach's results, Huynh, Mohar, Šámal, Thomassen and Wood [93] proposed an answer to Pach's question and sadly showed the following negative result.

Theorem 5.2 ([93])**.** *If U is a countable strongly-universal graph for the class of countable planar graphs, then the complete graph $K_{\aleph_0}$ is a minor of U.*

Finite universal graphs In this chapter, we focus on finite graphs[1]. Ulam's question, as well as Pach's follow-up question, can also be considered in a finite setting. Of course, for any class of finite graphs $\mathcal{G}$ that is closed under disjoint union, there exists a family $(U_n)_{n\in\mathbb{N}}$ of faithful strongly-universal graphs for $\mathcal{G}$. For example define U_n as the disjoint union of

[1]We assume that all graphs considered from now on are finite, unless stated otherwise

all n-vertex graphs in $\mathcal{G}$. However, this solution is far from satisfactory, as the size of U_n grows sharply[2] with n. A proper reformulation of Ulam's question could be:

Problem 5.3. *Does there exist a family of graphs $(U_n)_{n\in\mathbb{N}}$ faithful strongly-universal for the class of planar graphs with $|U_n| = n^{O(1)}$?*

Sadly, Bergold, Iršič, Lauff, Orthaber, Scheucher and Wesolek recently gave a negative answer to this question [33]. Once again, a similar question to the one Pach raised in the infinite case was formulated in [33]: Does a universal graph satisfying Problem 5.3 exist if we allow a relaxation of the faithful condition? We show, similarly to Huynh, Mohar, Šámal, Thomassen, and Wood [93], that even when the "faithful" condition is weakened to "H-minor-free", such a universal graph still does not exist.

Theorem 5.4. *There is a polynomial function $f_{5.4}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let $t, \ell \geqslant 2$ be integers. If U is a K_t-minor-free graph containing every $\ell \times \ell$ triangulated grid as subgraph, then*

$$|V(U)| \geqslant 2^{\frac{1}{f_{5.4}(t)}\cdot \ell}.$$

In particular, for every integer $t \geqslant 5$ there exists a constant $C_t > 0$ such that for every integer $n \geqslant 2$, every K_t-minor-free graph containing every n-vertex planar graph as subgraph has at least $2^{C_t\sqrt{n}}$ vertices.

The question first raised by Pach in [148] about universal graphs for the class of planar graphs can be naturally generalised to multiple other classes of graphs. Very little is known about faithful universal graphs in the finite setting. Remember that, as explained earlier, all the techniques generally used to construct universal graphs in this setting do not produce faithful or "close" to faithful universal graphs. Moreover, in the few cases where some non-trivial faithful universal graphs are known, their size is significantly larger than that of the optimal universal graphs. Moreover, in the few cases where some non-trivial faithful universal graphs were known, their size was significantly larger than the size of the optimal universal graphs. Consider for instance the case of trees. It was proved by Chung and Graham [55] that there is a weakly-universal graph with $\Theta(n \log n)$ edges for trees, and that this bound is best possible. As explained above, the proof uses separators and produces graphs with cliques of logarithmic size, so the resulting graphs are very far from trees (or any other class of sparse graphs). On the other hand, Gol'dberg and Livshits [83] constructed faithful weakly-universal graphs for trees on $2^{O(\log^2 n)}$ vertices, and this order of magnitude was later shown to be asymptotically best possible [56]. With Esperet, Groenland, Hilaire, Rambaud

[2] Similarly, in the infinite countable case, taking the disjoint union over $\mathcal{G}$ also yields a strongly-universal graph, but an uncountable one.

and Wesolek, we studied the existence of faithful or nearly faithful universal graphs for multiple natural graph parameters (i.e. treewidth, pathwidth, treedepth). These results are not included in this manuscript but are summarised in the two tables below. Note that the lower bounds for weakly-universal transfer directly to strongly-universal, and conversely, the upper bounds for strongly-universal transfer to weakly-universal, such results are mentioned only in one of the two tables, to improve readability.

Class	Exact	Approximation	
pathwidth $k \geqslant 2$	$2^{\Theta(n \log k)}$	pathwidth k^2	$n^{O_k(1)}$
treewidth $k \geqslant 2$	$2^{\Omega(n \log k)}$	treewidth $3k-1$	$n^{O_k(1)}$
K_4-minor-free	$2^{\Omega(n)}$	K_7-minor-free	$n^{O(1)}$
		K_5-minor-free	$n^{O(\log n)}$
K_t-minor-free, $t \in [5,6,7]$	$2^{\Omega(n/t)}$	$K_{t'}$-minor-free	$2^{\Omega_t(\sqrt{n})}$
K_t-minor-free, $t \geqslant 8$	$2^{\Omega(n/t)}$	$K_{t'}$-minor-free	$2^{\Omega_t(n)}$

Figure 5.1: Results for **weakly-universal** graphs.

Class	Exact	Approximation	
treedepth $k \geqslant 1$	$O(n^k)$	–	–
pathwidth $k \geqslant 2$	$2^{\Omega(n \log k)}$	pathwidth 2^k	$n^{O_k(1)}$
treewidth $k \geqslant 1$	$2^{\Omega(n \log k)}$	treewidth $3k-1$	$n^{O_k(1)}$
K_4-minor-free	$2^{\Omega_k(n)}$	K_7-minor-free	$n^{O(1)}$

Figure 5.2: Results for **strongly-universal** graphs.

Universal Poset The definition of universality extends naturally beyond graphs and can be defined for posets. A family of posets $(U_n)_{n\in\mathbb{N}}$ is said to be faithful strongly-universal for a class of posets $\mathcal{P}$ if, for any n-element poset $P \in \mathcal{P}$, P is an induced subposet of U_n. The topic of (countable) universal posets has a history even older than that of universal graphs [75, 99, 98, 74] and, as mentioned in [92], has motivated the research area of category theory. Nonetheless, the following natural question is still open:

Problem 5.5. *What is the minimum size of a poset U_n containing all n-element posets as an induced subposet?*

This problem was also raised in [38]. In this paper, Bonamy, Esperet, Groenland, and Scott, considered the systematic study of strongly-universal graphs for dense classes of graphs. They proved that classes of graphs that grow sufficiently quickly admit a universal graph of asymptotically optimal size.

Theorem 5.6 ([38])**.** *For $\alpha > 0$ any class of graphs $\mathcal{G}$ such that $|\mathcal{G}_n| \leqslant 2^{\alpha n^2}$ where $\mathcal{G}_n$ is the set containing all n-vertex graph of G, there exists a universal graphs of size $2^{(\alpha+o(1))n}$, and this is optimal up to the $2^{o(n)}$ factor.*

Using Theorem 5.6, Bonamy, Esperet, Groenland and Scott proved that the class of all posets admits an asymptotically optimal *comparability labeling scheme*, which translated to the following corollary in our setting.

Corollary 5.7 ([38])**.** *For any $n \in \mathbb{N}$, there exists a directed graph G with $|V(G)| \leqslant 2^{(1/4+o(1))n}$ such that, for any n-element poset P, the comparability graph of P is an induced subgraph of G.*

While this result is optimal in size up to the $2^{o(n)}$ factor, crucially, in Corollary 5.7, the graph G obtained is **not** the comparability graph of a poset, which is why they raised Problem 5.5. The following theorem makes a step toward answering the question.

Theorem 5.8. *For any $n \in \mathbb{N}$, there exists a poset U_n containing all n-element posets as an induced subposet, and $|U_n| \leqslant 2^{\frac{2}{3}n+O(\sqrt{n})}$.*

Roadmap We first give some preliminary definitions and tools in Section 5.2. In Section 5.3 we give a proof of Theorem 5.4 and in Section 5.3 we prove Theorem 5.8.

5.2 Preliminaries

VC-dimension We recall here an important theorem of set theory, about VC-dimension independently proven by Sauer and Shelah.

Given a set X, we write 2^X for the power set of X. Let $\mathcal{F} \subseteq 2^X$ be a given set system on the ground set X. The *VC-dimension* of $\mathcal{F}$ is the supremum taken over the integers $d \geqslant 0$ for which there is a set $S \subseteq X$ of size d which is shattered, that is,

$$\{Y \cap S : Y \in \mathcal{F}\} = 2^S.$$

Lemma 3.23. *[Sauer-Shelah lemma [166, 168]] If $\mathcal{F} \subseteq 2^{[n]}$ has VC-dimension d then,*

$$|\mathcal{F}| \leqslant \sum_{i=0}^{d} \binom{n}{i}.$$

For $n \geqslant 2d$ and d sufficiently large,

$$\sum_{i=0}^{d} \binom{n}{i} \leqslant n^d/(d!) + dn^{d-1}/((d-1)!) \leqslant n^d.$$

The upper bound $\leqslant n^d + dn^{d-1} \leqslant 2n^d$ is easy to see for all $n \geqslant d$ and the VC dimension d is always at most n.

Minors and edge count The following theorem will be useful in establishing the existence of a K_t minor in universal graphs.

Lemma 5.9 ([124, 176])**.** *There exists a polynomial function $f_{5.12} : \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let t be a positive integer. For every graph G, if K_t is not a minor of G, then*

$$|E(G)| \leqslant f_{5.12}(t)|V(G)|.$$

Integer partition We will encounter the following function while constructing universal posets. Let us define $p(n)$ as the number of ways we can *partition* an integer into smaller integers. Formally, $p(n)$ is the number of distinct non-increasing sequences $\lambda_1, \lambda_2, \ldots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i = n$.

Theorem 5.10. *The number of partitions of an integer, $p(n)$, as n tends to infinity, satisfies*

$$p(n) \sim \frac{1}{4\sqrt{3}n} \exp\left(\pi\sqrt{\frac{2n}{3}}\right).$$

5.3 Faithful universal graphs for minor-closed classes

This section focuses on proving Theorem 5.4, which gives a lower bound of $2^{\Omega_t(\sqrt{n})}$ on the order of a K_t-minor-free graph containing all planar n-vertex graphs as subgraph; and Theorem 5.16, which gives a lower bound of $2^{\Omega_t(n)}$ on the order of a K_t-minor-free graph containing all toroidal n-vertex graphs as subgraph. In both results, the implicit multiplicative constant in the exponent is a polynomial in t.

Theorem 5.4. *There is a polynomial function $f_{5.4}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let $t, \ell \geqslant 2$ be integers. If U is a K_t-minor-free graph containing every $\ell \times \ell$ triangulated grid as subgraph, then*

$$|V(U)| \geqslant 2^{\frac{1}{f_{5.4}(t)}\cdot \ell}.$$

In particular, for every integer $t \geqslant 5$ there exists a constant $C_t > 0$ such that for every integer $n \geqslant 2$, every K_t-minor-free graph containing every n-vertex planar graph as subgraph has at least $2^{C_t\sqrt{n}}$ vertices.

The main results in this section are Theorem 5.4, which gives a lower bound of $2^{\Omega_t(\sqrt{n})}$ on the order of a K_t-minor-free graph containing all planar n-vertex graphs as subgraph; and Theorem 5.16, which gives a lower bound of $2^{\Omega_t(n)}$ on the order of a K_t-minor-free graph containing all toroidal n-vertex graphs as subgraph. In both results, the implicit multiplicative constant in the exponent is a polynomial in t.

5.3.1 K_t-minors in grid with jumps

This subsection presents Lemma 5.11 stated below. Informally, this lemma allows us to construct clique minors in a triangulated grid with a few extra edges. Note that we can assume that $t \geqslant 5$, as the 3×3 grid, which is planar, has maximum degree at most 4 and has treewidth 3, contains a K_4-minor.

Let us first introduce some notation. For every positive integers a, b, the $a \times b$ *grid* is the graph with vertex set $[a] \times [b]$ and edge set $\{(x,y)(x,y+1) \mid x \in [a], y \in [b-1]\} \cup \{(x,y)(x+1,y) \mid x \in [a-1], y \in [b]\}$. For every $i \in [b]$, the i^{th} row of this grid is the set of vertices $\{(x,i) \mid x \in [a]\}$, and for every $j \in [a]$, the j^{th} column of this grid is the set $\{(j,y) \mid y \in [b]\}$.[3]

The *toroidal* $a \times b$ grid is obtained from the $(a+1) \times (b+1)$ grid by identifying the first column with the $(a+1)$-th column, and the first row with the $(b+1)$-row. Equivalently, the x-coordinates are considered modulo a and the y-coordinates are considered modulo b.

A *triangulated* $a \times b$ *grid* is a spanning supergraph G of the $a \times b$ grid such that for every $x \in [a-1], y \in [b-1]$, exactly one of the pairs $(x,y)(x+1,y+1)$ and $(x+1,y)(x,y+1)$ is an edge of G. A *triangulated toroidal* $a \times b$ *grid* is defined similarly, starting with the toroidal $a \times b$ grid instead of the $a \times b$ grid.

If G is a triangulated $a \times b$ grid, we write $R_i(G) = \{(x,i) : x \in [a]\}$ for the vertices in the ith row and $C_j(G) = \{(j,y) : y \in [b]\}$ for vertices in the jth column. When G is clear from the context, we will simply write R_i for $R_i(G)$ and C_j for $C_j(G)$.

[3] The coordinates are chosen so that rows and columns can be imagined in an $xy-$coordinate system.

A *jump*[4] in a triangulated grid G is a non-edge in G (i.e., a pair of non-adjacent vertices). Our main goal is to show that adding the edges corresponding to sufficiently many jumps to a triangulated grid will create a large complete minor. Since additional jumps near the boundary of the grid are less helpful, we will only be interested in jumps between vertices that are not too close to the boundary.

For integers i, t, ℓ, ℓ', we call row R_i of a triangulated $\ell \times \ell'$ grid *t-internal* if $i \in \{t+1, t+2, \dots, \ell' - t\}$. Similarly, we call column C_j *t-internal* if $j \in \{t+1, t+2, \dots, \ell - t\}$. We call a vertex $v = (x, y)$ *t-internal* if its column and row are t-internal.

Given a triangulated grid G and a set of jumps M, we denote by $G \cup M$ the graph with vertex set $V(G)$ and edge set $E(G) \cup M$. We will need the following folklore lemma, whose proof is omitted here but can be found in [22].

Lemma 5.11. *There is a polynomial function $f_{5.11} \colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let t be a positive integer, let ℓ, ℓ' be integers with $\ell, \ell' \geqslant 2f_{5.11}(t)$. Let G be a triangulated $\ell \times \ell'$ grid. For every set M of pairwise disjoint jumps of G, if*

1. *for every $uv \in M$, u or v is $f_{5.11}(t)$-internal; and*
2. *$|M| \geqslant f_{5.11}(t)$;*

then K_t is a minor of $G \cup M$.

5.3.2 Creating the jumps

Together with Lemma 5.11 above, we will need the following two classical results.

Lemma 5.12 ([124, 176])**.** *There is a polynomial function $f_{5.12} \colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let t be a positive integer. For every graph G, if K_t is not a minor of G, then*

$$|E(G)| \leqslant f_{5.12}(t)|V(G)|.$$

Given a set X, we write 2^X for the power set of X. Let $\mathcal{F} \subseteq 2^X$ be a given set system on the ground set X. The *VC-dimension* of $\mathcal{F}$ is the supremum taken over the integers $d \geqslant 0$ for which there is a set $S \subseteq X$ of size d which is *shattered*, that is such that $\{Y \cap S : Y \in \mathcal{F}\} = 2^S$.

Lemma 5.13 (Sauer-Shelah [167, 169])**.** *If the VC-dimension of $\mathcal{F} \subseteq 2^{[n]}$ is at most d, then $|\mathcal{F}| \leqslant \sum_{i=0}^{d} \binom{n}{i} \leqslant 2n^d$.*

[4] Note that our definition of a jump is slightly different from the definition of a jump in [93], but the spirit is the same.

We will combine Lemma 5.11 with Lemmas 5.12 and 5.13 above to prove the following result, which informally says that a K_t-minor-free graph on ℓ^2 vertices cannot contain too many triangulated $\ell \times \ell$ grids as a (spanning) subgraph.

Lemma 5.14. *There is a polynomial function $f_{5.14}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let $t, \ell \geqslant 2$ be integers. Let U be a K_t-minor-free graph on ℓ^2 vertices. Then there are at most $2^{f_{5.14}(t)\cdot \ell \log \ell}$ edge sets $S \subseteq E(U)$ such that $U[S]$ induces a triangulated $\ell \times \ell$ grid.*

Proof. Let

$$d = 15(4f_{5.12}(t)f_{5.11}(t) + f_{5.11}(t)) \cdot \ell.$$

Choose $f_{5.14}(t) = \Theta(d/\ell)$ (i.e. a sufficiently large constant times d/ℓ), such that for any $t, \ell \geqslant 2$,

$$2^{f_{5.14}(t)\cdot \ell \log \ell} > 2(f_{5.12}(t) \cdot \ell^2)^d.$$

Let U be a K_t-minor-free graph on $N = \ell^2$ vertices. Suppose towards a contradiction that U has more than $2^{f_{5.14}(t)\cdot \ell \log \ell}$ edge sets inducing a triangulated $\ell \times \ell$ grid.

Let $\mathcal{G} \subseteq 2^{E(U)}$ be the family of these edge sets. By Lemma 5.12, $|E(U)| \leqslant f_{5.12}(t) \cdot \ell^2$. Since

$$|\mathcal{G}| > 2^{f_{5.14}(t)\cdot \ell \log \ell} > 2\left(f_{5.12}(t) \cdot \ell^2\right)^d \geqslant 2|E(U)|^d,$$

by Lemma 5.13, there exists $A \subseteq E(U)$ with $|A| = d + 1$ such that for every $B \subseteq A$, there exists $E_B \in \mathcal{G}$ such that $E_B \cap A = B$. In particular, E_A and $E_\emptyset$ belong to $\mathcal{G}$, so the graph $G_1 = U[A]$ is a subgraph of a triangulated $\ell \times \ell$ grid and the graph $G_2 = U[E_\emptyset]$ is a triangulated $\ell \times \ell$ grid.[5] Since G_1 has maximum degree at most 8, there exists a matching M_0 included in A of size at least $\frac{|A|}{15} \geqslant (4f_{5.12}(t)f_{5.11}(t) + f_{5.11}(t)) \cdot \ell$.

Let C be the set of $f_{5.11}(t)$-internal vertices of G_2. Let M_1 be the set of all the pairs in M_0 intersecting C. Note that since $M_1 \subseteq A$, no edge in M_1 is an edge of G_2, so M_1 is a set of pairwise disjoints jumps between $f_{5.11}(t)$-internal vertices of G_2. We now show that $|M_1| \geqslant f_{5.11}(t)$.

Note that $U[M_0 \setminus M_1]$ is a subgraph of the K_t-minor-free graph U with vertex set included in $V(U) \setminus C$, and $|V(U) \setminus C| \leqslant 4f_{5.11}(t) \cdot \ell$. It follows that by Lemma 5.12,

$$|M_0 \setminus M_1| \leqslant f_{5.12}(t) \cdot |V(U) \setminus C| \leqslant 4f_{5.12}(t)f_{5.11}(t) \cdot \ell.$$

Therefore, $|M_1| = |M_0| - |M_0 \setminus M_1| \geqslant f_{5.11}(t)$, and we conclude by Lemma 5.11 that K_t is a minor of $G_2 \cup M_1$, and so of U, yielding the desired contradiction. □

We are now ready to prove the main result of this section.

[5] We note that we do not need the full power of the Sauer-Shelah lemma here, just a simple consequence on set systems in which the difference between any two sets has size at most d. However, the full power of the Sauer-Shelah lemma will be needed later.

Theorem 5.4. *There is a polynomial function $f_{5.4}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let $t, \ell \geqslant 2$ be integers. If U is a K_t-minor-free graph containing every $\ell \times \ell$ triangulated grid as subgraph, then*

$$|V(U)| \geqslant 2^{\frac{1}{f_{5.4}(t)}\cdot \ell}.$$

In particular, for every integer $t \geqslant 5$ there exists a constant $C_t > 0$ such that for every integer $n \geqslant 2$, every K_t-minor-free graph containing every n-vertex planar graph as subgraph has at least $2^{C_t\sqrt{n}}$ vertices.

Proof. Let $\ell \geqslant 2$ be an integer and let U be a K_t-minor-free graph containing every triangulated $\ell \times \ell$ grid as subgraph. Note that in particular, $|V(U)| \geqslant \ell^2 \geqslant 2$. We define

$$d = 120\ell \cdot f_{5.12}(t) f_{5.11}(t) + 48,$$

and let $f_{5.4}(t) = \Theta((f_{5.14}(t))^2 \cdot d/\ell)$ (i.e., a sufficiently large constant times $(f_{5.14}(t))^2 \cdot d/\ell$). We assume towards a contradiction that $N = |V(U)| \leqslant 2^{\frac{1}{f_{5.4}(t)}\cdot \ell}$. Since $N \geqslant 2$, we can assume that $\ell \geqslant f_{5.4}(t)$.

Given an $\ell \times \ell$ grid G with vertex set $[\ell^2]$, there are at least $2^{(\ell-1)^2-4}$ spanning supergraphs of G that are isomorphic to a triangulated $\ell \times \ell$ grid in which the 4 corners have degree 2. Any such triangulated grid has at most 24 automorphisms, since once you fix the image of the degree two vertices, the remaining vertices are determined by their distances to these vertices. This shows that in order to contain all $\ell \times \ell$-triangulated grids as subgraph, the graph U needs to have at least $\frac{1}{24}2^{(\ell-1)^2-4} \geqslant 2^{(\ell-1)^2-9}$ distinct edge sets inducing such a triangulated grid. By Lemma 5.14, for every subset $A \subseteq V(U)$ of size ℓ^2, at most $2^{f_{5.14}(t)\cdot \ell \log \ell}$ triangulated $\ell \times \ell$ grids can be accounted for by $U[A]$. Hence there is a family $\mathcal{G} \subseteq 2^{V(U)}$ of at least

$$2^{(\ell-1)^2-9-f_{5.14}(t)\cdot \ell \log \ell} > 2^{\ell^2/2+1} \geqslant 2\left(2^{\frac{1}{f_{5.4}(t)}\cdot \ell}\right)^d \geqslant 2N^d \tag{5.1}$$

vertex sets of size ℓ^2 of U that contain an $\ell \times \ell$ triangulated grid. In the computation above we have used the fact that $\ell \geqslant f_{5.4}(t) = \Theta((f_{5.14}(t))^2)$ in the first inequality (so that $(\ell-1)^2 - 9 - f_{5.14}(t)\cdot \ell \log \ell \geqslant (\ell-1)^2 - \ell^{3/2}\log \ell \geqslant \ell^2/2 + 1$ for sufficiently large ℓ), and the fact that $f_{5.4}(t) = \Theta(d/\ell)$ in the second inequality.

It then follows from Lemma 5.13 that the VC-dimension of $\mathcal{G}$ is at least $d+1$. Hence there exists $A \subseteq V(U)$ of size $d+1$ such that for every $B \subseteq A$, there exists a $V_B \in \mathcal{G}$ such that

$$V_B \cap A = B.$$

In particular, $A \subseteq V_A \in \mathcal{G}$. Let G_1 be a triangulated $\ell \times \ell$ grid spanning $U[V_A]$.

Note that the vertices of every triangulated grid can be covered by the following four (not necessarily disjoint) induced paths: two horizontal paths covering the top and bottom rows of the grid, one path containing all vertices in the odd numbered columns (except for possibly those on the top and bottom rows) and the other for the vertices in the even numbered columns. By the pigeonhole principle, this means that G_1 contains an induced path P containing at least $|A|/4$ vertices from A.

We select $k = \lfloor |V(P)|/3 \rfloor \geqslant |A|/12 - 1$ vertex-disjoint subpaths $P_1, \ldots, P_k$ of P such that for each $1 \leqslant i \leqslant k$, the two endpoints of P_i are in A and exactly one internal vertex of P_i lies in A (so each P_i contains precisely 3 vertices of A). Let $B \subseteq A$ be the set of endpoints of the paths P_i, $1 \leqslant i \leqslant k$. Note that every path P_i contains a vertex of $A \setminus B$.

We then consider a set $V_B \in \mathcal{G}$ such that $V_B \cap A = B$ and let G_2 be a triangulated $\ell \times \ell$ grid spanning $U[V_B]$. We will show that $G_1 \cup G_2$ contains a K_t-minor, hence contradicting the fact that U is K_t-minor-free.

Let us construct two families of paths $\mathcal{Q}_1$ and $\mathcal{Q}_2$ satisfying the following properties:

1. the members of $\mathcal{Q}_1$ are pairwise disjoint paths in G_2 internally disjoint from $V(G_1)$, and whose endpoints are (pairwise disjoint) jumps of G_1;
2. the members of $\mathcal{Q}_2$ are pairwise disjoint paths in G_1 internally disjoint from $V(G_2)$, and whose endpoints are (pairwise disjoint) jumps of G_2.

For any $1 \leqslant i \leqslant k$, we do the following. By definition, P_i intersects $A \setminus B$, and thus P_i contains a subpath Q_i between two vertices $x, y \in V(G_2)$ such that Q_i contains at least one internal vertex, and such that all internal vertices of Q lie in $V(G_1) \setminus V(G_2)$. If the endpoints x, y of Q_i are non-adjacent in G_2, then add Q_i to $\mathcal{Q}_2$. Otherwise add the path consisting of the single edge xy to $\mathcal{Q}_1$. See Figure 5.3 for an illustration. Observe that the two properties above are maintained in both cases.

Since $\mathcal{Q}_1 + \mathcal{Q}_2 = k \geqslant |A|/12 - 1$, there exists $a \in \{1, 2\}$ such that $|\mathcal{Q}_a| \geqslant |A|/24 - 2 \geqslant d/24 - 2$. Fix such $a \in \{1, 2\}$, and let M_0 be the family of all the pairs of endpoints of paths in $\mathcal{Q}_a$. Since $\mathcal{Q}_a$ is a family of pairwise disjoint paths internally disjoint from $V(G_a)$, $G_a \cup M_0$ is a minor of U. Moreover, M_0 is disjoint from $E(G_a)$ and

$$|M_0| = |\mathcal{Q}_a| \geqslant d/24 - 2 = \tfrac{1}{24}\big(120 f_{5.12}(t) f_{5.11}(t) \cdot \ell + 48\big) - 2 \geqslant 5 f_{5.12}(t) f_{5.11}(t) \cdot \ell.$$

Let C be the set of $f_{5.11}(t)$-internal vertices of G_a, and let M_1 be the set of all pairs in M_0 with at least one endpoint in C.

Consider the graph H with vertex set $V(G_a) \setminus C$ and edge set $M_0 \setminus M_1$. Then, by Lemma 5.12, since H is a minor of U, $|E(H)| \leqslant f_{5.12}(t)|V(H)|$. This implies that

$$|M_0 \setminus M_1| \leqslant f_{5.12}(t) \cdot |V(G_a) \setminus C| \leqslant f_{5.12}(t) \cdot 4\ell f_{5.11}(t),$$

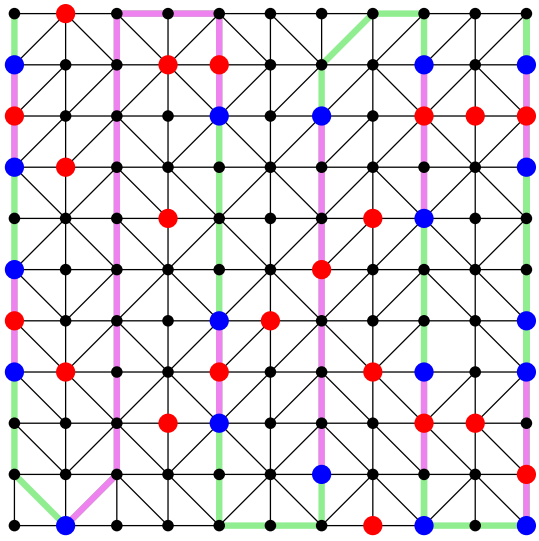


Figure 5.3: Illustration for the proof of Theorem 5.4. The depicted grid is G_1, the red and blue vertices represent the shattered set A, and the blue vertices are the members of the subset B. Along a path P in the grid (in bold green and violet) we build disjoint subpaths of P between vertices in B via a vertex in $A \setminus B$. These subpaths are depicted in violet. This ensures that every such path contains a path in $V(G_1)$ between two vertices $u, v \in V(G_2)$ of length at least 2 internally disjoint from $V(G_2)$. If u, v are non-adjacent in G_2, then this creates a jump in G_2, and we put it in $\mathcal{Q}_2$. Otherwise, the edge uv creates a jump in G_1, that we put in $\mathcal{Q}_1$. At the end of this process, either $\mathcal{Q}_1$ or $\mathcal{Q}_2$ is large, which will imply the existence of a K_t-minor.

and so

$$|M_1| \geqslant |M_0| - 4\ell f_{5.11}(t) f_{5.12}(t) \geqslant f_{5.11}(t).$$

We conclude by Lemma 5.11 that K_t is a minor of $G_a \cup M_1$, and so of U. □

Considering toroidal grids instead of planar grids leads to significantly better lower bounds on the size of subgraph-universal graphs. This is due to the following variant of Lemma 5.11, which avoids boundary conditions on the endpoints of jumps.

Lemma 5.15. *There is a polynomial function $f_{5.15} : \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ such that the following holds. Let t be a positive integer, let ℓ, ℓ' be integers with $\ell, \ell' \geqslant 2f_{5.15}(t)$. Let G be a triangulated toroidal $\ell \times \ell'$ grid. For every set M of pairwise disjoint jumps of G with $|M| \geqslant f_{5.15}(t)$, K_t is a minor of $G \cup M$.*

Proof. Consider a random spanning triangular planar $\ell \times \ell'$ subgrid G' of G, obtained by selecting the bottom left corner of G' uniformly at random in G, and let $k \leqslant \min\{\ell/4, \ell'/4\}$. Note that the expected number of jumps from M having at least one k-internal endpoint in G' is at least

$$\frac{(\ell - 2k)(\ell' - 2k)}{\ell\ell'} \cdot |M| \geqslant \frac{|M|}{4}.$$

It follows that G has a planar triangular $\ell \times \ell'$ subgrid G' and M has a subset M' of at least $\frac{1}{4}|M|$ jumps in G', each having at least one k-internal endpoint. Setting $f_{5.15}(t) = 4f_{5.11}(t)$, we can apply Lemma 5.11 and obtain a K_t-minor in $G' \cup M'$, and thus also in $G \cup M$. □

So while in the planar case we needed $\Omega_t(\ell)$ jumps to produce a K_t-minor, in the toroidal case we only need a constant number $O_t(1)$ of jumps to produce a K_t-minor.

A slight adjustment to the calculations in the proofs above, simply replacing applications of Lemma 5.11 by applications of Lemma 5.15, gives the following result (as the proof is mostly the same we only sketch the argument).

Theorem 5.16. *There exists a polynomial function $p(t)$ such the following holds. Let t, ℓ be positive integers with $\ell \geqslant 2$.*

- *Let U be a K_t-minor-free graph on ℓ^2 vertices. Then there are at most $\ell^{p(t)}$ edge sets $S \subseteq E(U)$ such that $U[S]$ induces a triangulated toroidal $\ell \times \ell$ grid.*
- *Let U be a K_t-minor-free graph containing every triangulated toroidal $\ell \times \ell$ grid as subgraph. Then $|V(U)| \geqslant 2^{\ell^2/p(t)}$.*

In particular, for every integer $t \geqslant 8$ there exists a constant $C_t > 0$ such that for every positive integer n, every K_t-minor-free graph containing every n-vertex K_8-minor-free graph as subgraph has at least $2^{C_t n}$ vertices.

Proof sketch. For the first item, we closely follow the proof of Lemma 5.14. We start by defining $d_1 = 15 \cdot f_{5.15}(t)$.

Let $\mathcal{G} \subseteq 2^{E(U)}$ be the family of edge sets of U inducing a triangulated toroidal $\ell \times \ell$ grid. We define $p_1(t)$ such that

$$\ell^{p_1(t)} > 2(f_{5.12}(t)\ell^2)^{d_1}.$$

Assume for the sake of contradiction that $|\mathcal{G}| \geqslant \ell^{p_1(t)}$. By Lemma 5.12, $|E(U)| \leqslant f_{5.12}(t)\ell^2$ and thus $|\mathcal{G}| > 2|E(U)|^{d_1}$. By Lemma 5.13, there exists $A \subseteq E(U)$ with $|A| = d_1 + 1$ such that for every $B \subseteq A$, there exists $E_B \in \mathcal{G}$ such that $E_B \cap A = B$. Let $G_1 = U[A]$ and $G_2 = U[E_\emptyset]$. Since G_1 has maximum degree at most 8, there exists a matching M included in A of size at least $\frac{|A|}{15} \geqslant f_{5.15}(t)$. The set M corresponds to a set of pairwise disjoint jumps in G_2, and it thus follows from Lemma 5.15 that $G_2 \cup M$ contains a K_t-minor, and thus U also contains a K_t-minor, which is a contradiction.

For the second item, we follow closely the proof of Theorem 5.4. We set $d_2 = 24 f_{5.15}(t) + 48$, and $p_2(t) = \Theta(d_2)$ (i.e., some sufficiently large constant times d_2). Let U be a K_t-minor-free graph containing all triangular toroidal $\ell \times \ell$ grids, and assume for the sake of contradiction that $N = |V(U)| < 2^{\ell^2/p_2(t)}$. There are $2^{\Omega(\ell^2)}$ non-isomorphic triangular toroidal $\ell \times \ell$ grids, and for each set $A \subseteq V(U)$ of size ℓ^2, at most $2^{p_1(t)\log \ell}$ triangular toroidal $\ell \times \ell$ grids can be accounted for by $U[A]$. Hence there is a family $\mathcal{G} \subseteq 2^{V(U)}$ of $2^{\Omega(\ell^2) - p_1(t)\log \ell}$ vertex subsets of size ℓ^2 of U that contain a triangulated toroidal n $\ell \times \ell$ grid.

We obtain

$$|\mathcal{G}| = 2^{\Omega(\ell^2) - f_1(t)\cdot \ell \log \ell} > 2\left(2^{\ell^2/p_2(t)}\right)^{d_2} > 2N^{d_2},$$

and thus the VC-dimension of $\mathcal{G}$ is at least $d_2 + 1$ by Lemma 5.13. Hence there exists $A \subseteq V(U)$ of size $d_2 + 1$ such that for every $B \subseteq A$, there exists a $V_B \in \mathcal{G}$ such that $V_B \cap A = B$.

Let G_1 be a toroidal $\ell \times \ell$-grid spanned by $U[V_A]$. Using an induced path in G_1 covering at least $|A|/4$ vertices of A, we choose at least $|A|/12 - 2$ induced paths $P_1, \ldots, P_k$ in G_1 whose endpoints are both in A and with exactly one internal vertex in A, and we call B the subset of A consisting of the endpoints of the paths P_i.

We then select a subset V_B of $V(U)$ such that $U[V_B]$ spans a triangular toroidal $\ell \times \ell$-grid G_2, and for which $V_B \cap A = B$. Using the paths P_i, we create at least $|A|/24 - 2 \geqslant f_{5.15}(t)$ pairwise disjoint jumps either in G_1 or G_2. It then follows from Lemma 5.15 that U contains K_t as a minor, which is a contradiction. □

5.4 Strongly-universal poset

This section aims to prove the following theorem.

Theorem 5.8. *For any $n \in \mathbb{N}$, there exists a poset U_n containing all n-element posets as an induced subposet, and $|U_n| \leqslant 2^{\frac{2}{3}n+O(\sqrt{n})}$.*

The proof uses the concept of *labeling schemes*, a well-known tool for designing universal graphs. The elements of the universal poset U_n will be all the possible "labels" in some cleverly chosen set $\mathcal{L}$ of labels, that is, $U_n = (\mathcal{L}, \preccurlyeq)$ for some relation function $\preccurlyeq$ to be defined. The relation $\preccurlyeq$ will be constructed such that for any two elements $\ell_1, \ell_2 \in \mathcal{L}$, the relation between ℓ_1 and ℓ_2 through $\preccurlyeq$ is a function of only the two labels ℓ_1, ℓ_2. With this terminology, proving that U is a universal poset boils down to proving the following two properties about $(\mathcal{L}, \preccurlyeq)$:

- $\preccurlyeq$ is a partial order (i.e transitive, reflexive and antisymmetric)
- For any n-element poset P, there exists a set of n labels $\{\ell_1, \ldots, \ell_n\}$ such that $(\{\ell_1, \ldots, \ell_n\}, \preccurlyeq)$ is isomorphic to P.

In the proof of Theorem 5.8, the universal poset U_n is constructed as the disjoint union of multiple smaller universal posets. These posets can be grouped into two categories, corresponding to Lemma 5.17 and Lemma 5.18, respectively. Lemma 5.17 constructs universal structures for posets containing a large antichain, while Lemma 5.18 handles posets with long chains.

Lemma 5.17. *For any $n, a, b \in \mathbb{N}$, such that $a + b \leqslant n$, let $\mathcal{Q}_n(a, b)$ be the family of all n-element posets P which contain an antichain of size a, and there are exactly b elements of P which are strictly smaller than some element in A.*

There exists a poset $Q_n(a, b)$ strongly-universal for $\mathcal{Q}(n, a, b)$ on $n2^{n-a}$ elements.

Lemma 5.18. *For any $n \in \mathbb{N}$ and $m_1, \ldots, m_n \geqslant 0$, let $\mathcal{S}_n(m_1, \ldots, m_n)$ denote the family of all n-element posets P with a chain decomposition with exactly m_j chains of length j for each $j \in [n]$. Let $a = \sum_{i=1}^{n} m_i$.*

There exists a poset $S_n(m_1, \ldots, m_n)$ strongly-universal for $\mathcal{S}_n(m_1 \ \ldots, m_n)$ on $na(\frac{n}{a}+1)^a$ elements.

Let us first prove Theorem 5.8 from the two lemmas stated above.

Proof of Theorem 5.8. Let $\mathcal{P}_n$ denote the family of all n-element poset and let $Q_n(a, b)$ (resp. $S_n(m_1, \ldots, m_n)$) be the poset defined in Lemma 5.17 (resp. Lemma 5.18). Dilworth's theorem Theorem 0.2 implies that every poset P has either an antichain of size $n/3$ or a chain

decomposition using at most $n/3$ chains. Therefore the poset U_n defined as the disjoint union of all posets in the following set is strongly-universal for $\mathcal{P}_n$,

$$\left\{ Q_n(a,b) \;\middle|\; a \in \left[\frac{n}{3}, n\right], b \in \mathbb{N} \right\} \cup \left\{ S_n(m_1, \dots, m_n) \;\middle|\; \sum_{i=1}^{n} m_i \leqslant \frac{n}{3} \right\}.$$

Let us now discuss the size of U_n.

$$\begin{aligned}
|U_n| &\leqslant \left(\sum_{a=n/3}^{n} \sum_{b=0}^{n-a} |Q(a,b)| \right) + \left(\sum_{\substack{m_1,\dots,m_n \in \mathbb{N} \\ m_1+\dots+m_n \leqslant n/3}} |S_n(m_1 \dots, m_n)| \right) \\
&\leqslant n^2 \cdot n 2^{2n/3} + p(n) \cdot na \left(\frac{n}{n/3} + 1 \right)^{n/3} \\
&\leqslant n^3 2^{2n/3} + 2^{O(\sqrt{n})} n^2 4^{n/3} \\
&= 2^{\frac{2}{3}n + O(\sqrt{n})}
\end{aligned}$$

Where $p(n)$ is the number of partition of the integer n, and the third inequality follows from Theorem 5.10. □

Let us now prove Lemma 5.17 and Lemma 5.18, both of which rely on the labeling method described earlier.

Proof of Lemma 5.17. The ground set of $Q_n(a,b)$ is all possible triples (x, L_x, H_x) where $x \in [n]$, $L_x \subseteq \{1, \dots, x-1\} \setminus \{b+1, \dots, b+a+1\}$, and $H_x \subseteq \{x+1, \dots, n\} \setminus \{b+1, \dots, b+a+1\}$. We set $(x, L_x, H_x) \prec (y, L_y, H_y)$ if and only if the following holds:

(1) $x < y$,

(2) $\{x, y\} \not\subseteq \{b+1, \dots, b+a\}$,

(3) $L_x \subseteq L_y$ and $H_x \supseteq H_y$,

(4) if $x \leqslant b$ then $x \in L_y$,

(5) if $y \geqslant b+a+1$ then $y \in H_x$.

We also set $(x, L_x, H_x) \preccurlyeq (x, L_x, H_x)$.

The intuition behind this labeling is as follows. Consider a poset $Q \in \mathcal{Q}_n(a,b)$, and a linear extension of Q with an antichain A of size a, starting at position $b+1$ and ending at

position $b+a+1$ in the linear extension. The label of an element $q \in Q$ will be (x, L_x, H_x), where x represents the position of the element in the linear extension described above. L_x (resp. H_x) represents the down-set (resp. up-set) of the element in $P \setminus A$. Crucially, q does not record its adjacency to the antichain A in its own label, allowing us to significantly reduce the label size. This is intuitively possible because if $q \preccurlyeq a$ in P with $q \notin A$ and $a \in A$, then the relation to q will be recorded in the label of a.

Let us now provide a formal proof that $(Q_n, \preccurlyeq)$ is a well-defined poset and that it is strongly-universal.

Claim 5.19. *$\preccurlyeq$ is a partial order on $Q_n(a, b)$.*

Proof. $\preccurlyeq$ is reflexive by definition. (1) ensures the antisymmetry. Let us now discuss the transitivity of $\preccurlyeq$. Suppose $(x, L_x, H_x) \prec (y, L_y, H_y)$ and $(y, L_y, H_y) \prec (z, L_z, H_z)$. Then $x < y < z$, $L_x \subseteq L_y \subseteq L_z$, and $H_x \supseteq H_y \supseteq H_z$. If $x \leqslant b$ then $x \in L_y$ and so $x \in L_z$. Similarly, if $y \geqslant b + a + 1$ then $z \in H_y$ and so $z \in H_x$. Therefore, $(x, L_x, H_x) \prec (z, L_z, H_z)$. Finally, it cannot be that $x, z \in \{b + 1, \dots, b + a\}$ as then $y \in \{b + 1, \dots, b + a\}$ and so we would not have $(x, L_x, H_x) \prec (y, L_y, H_y)$ due to (2). □

Claim 5.20. *Any poset $Q \in \mathcal{Q}_n(a, b)$ is an induced subposet of $(Q_n(a, b), \preccurlyeq)$.*

Proof. Let us consider a linear extension L of Q and denote by $L(x)$, for $x \in Q$, the number of elements in the closed down-set of x in L. Since $Q \in \mathcal{Q}_n(a, b)$, we can choose L such that the set $\{x \in Q \mid L(x) \in [b + 1, b + a + 1]\}$ forms an antichain in Q. To simplify notation and without loss of generality, we assume the ground set of Q to be $[n]$, where each element $x \in Q$ is now referred to as the element $L(x)$. In particular, under this new notation, $A := \{b+1, \dots, b+a\} \subseteq Q$ is an antichain. For each $x \in Q$, let $u_x = (x, L_x, H_x)$, where $L_x = \{x' \mid x' \prec x\} \setminus A$ and $H_x = \{x' \mid x \prec x'\} \setminus A$. Let us show that $\{u_x \mid x \in Q\}$ is an induced copy of Q in $(Q_n(a, b), \preccurlyeq)$.

Suppose $x \prec y$ in P. Then $x < y$ and $\{x, y\} \not\subseteq A$ since A is an antichain. Moreover, $L_x \subseteq L_y$ and $H_x \supseteq H_y$, therefore properties (1), (2), and (3) are satisfied. If $x \leqslant b$, then $x \notin A$ and so $x \in L_y$. Similarly, if $y \geqslant b + a + 1$, then $y \notin A$ and so $y \in H_x$. Therefore, properties (4) and (5) are also satisfied. Consequently, $u_x \prec u_y$ in $Q_n(a, b)$.

Suppose now that x and y are incomparable in Q. This means that $x \notin L_y$ and $y \notin H_x$. Without loss of generality, we may assume that $x < y$. If $\{x, y\} \subseteq A$, then by (2), we have that u_x and u_y are not comparable. Otherwise, we must have $x \leqslant b$ or $y \geqslant b+a+1$, in which case either (4) or (5) fails to hold. Thus, u_x and u_y are not comparable, which concludes the proof that $\{u_x \mid x \in Q\}$ forms an induced copy of Q.

□

Claim 5.19 and Claim 5.20 combined imply that $Q_n(a,b)$ is strongly-universal for $\mathcal{Q}_n(a,b)$. The size constraint follows from a simple count of the possible labels: there are n possible values for x, and $L_x \cup H_x$ can be encoded using 2^{n-a} bits, leading to the claimed upper bound on $|Q_n|$.

□

We are now left to prove Lemma 5.18.

Proof of Lemma 5.18. By definition, any poset S in $\mathcal{S}_n(m_1,\ldots,m_n)$ admits a chain decomposition with exactly m_k chains of size k. Let us consider the chains to be ordered in non-increasing order, and let ℓ_k denote the size of the k^{th} chain. We also denote by a the total number of chains, $a := \sum_{k\in[n]} m_k$. Note that the sequence $\ell_1,\ldots,\ell_a$ is the same for every $S \in \mathcal{S}_n(m_1,\ldots,m_n)$.

Let us define the ground set of $S_n(m_1,\ldots,m_n)$ as all possible tuples of the form $(x,c,i_1,\ldots,i_a)$, where $x \in [n]$, $c \in [a]$, and $i_k \in \{0,1,\ldots,\ell_k\}$ for each $k \in [a]$, with the additional constraint that $i_c \geqslant 1$. Intuitively, given a poset S that we want to embed and an element $s \in S$, x will once again denote the position in a linear extension of the poset we wish to embed. Given the chain decomposition satisfying the conditions mentioned above, where the chains are numbered in non-increasing order, c will denote the position of the chain containing s, and i_k will denote the largest element in the k^{th} chain smaller than s in S (or 0 if no such element exists).

Formally, we define $\preccurlyeq$ as follows: Given two labels $(x,c,i_1,\ldots,i_a)$ and $(y,d,j_1,\ldots,j_a)$, $(x,c,i_1,\ldots,i_a) \prec (y,d,j_1,\ldots,j_a)$ if and only if $x < y$ and $i_\ell \leqslant j_\ell$ for all $\ell \in [j]$.

Claim 5.21. *$\preccurlyeq$ is a partial order on $S_n(m_1,\ldots,m_n)$.*

Proof. $\preccurlyeq$ is reflexive by definition. If $(x,c,i_1,\ldots,i_a) \preccurlyeq (y,d,j_1,\ldots,j_a)$ and $(y,d,j_1,\ldots,j_a) \preccurlyeq (x,c,i_1,\ldots,i_a)$ then $x = y$ and therefore $(x,c,i_1,\ldots,i_a) = (y,d,j_1,\ldots,j_a)$, which ensures anti-symmetry.

It remains to check transitivity. Suppose,

$$(x,c,i_1,\ldots,i_a) \prec (y,d,j_1,\ldots,j_a) \qquad \text{and} \qquad (y,d,j_1,\ldots,j_a) \prec (z,e,h_1,\ldots,h_a).$$

Then $x < y < z$ and for each $k \in [a]$, $i_k \leqslant j_k \leqslant h_k$ which ensures,

$$(x,c,i_1,\ldots,i_a) \prec (z,e,h_1,\ldots,h_a).$$

□

Claim 5.22. *Any poset $S \in \mathcal{S}_n(m_1,\ldots,m_n)$ is an induced subposet of $(S_n(m_1,\ldots,m_n),\preccurlyeq)$.*

Proof. Consider $(S, \leqslant) \in \mathcal{S}_n(m_1, \dots, m_n)$, and consider a chain decomposition $\mathcal{C} = (C_1, \dots, C_a)$ such that C_k chain has length ℓ_k. Let us also fix a linear extension L of S and denote by $L(x)$, for $x \in S$, the number of elements in the closed down-set of x in L.

Let x be an element of S in chain C_c, for each $k \in [a]$, consider $x_1 \leqslant x_2 \leqslant \dots \leqslant x_{\ell_k}$ to denote the element of C_k. Let $i_k(x)$ be the largest integer i such that $x_i \leqslant x$ and fix $i = 0$ if no such integer x_i exists. Since x is in C_c, we have $i_c(x) \geqslant 1$, therefore we can associate to x the label $\mathrm{lab}(x) = (L(x), c, i_1(x), \dots, i_a(x)) \in S_n(m_1, \dots, m_n)$.

We show that $\{\mathrm{lab}(x) : x \in S\}$ forms an induced copy of S in $S_n(m_1, \dots, m_n)$. First, consider two elements $x, y \in S$ such that $x < y$, then $L(x) < L(y)$ as L is a linear extension. Moreover, for any $k \in [a]$, if $i_k(x) > 0$, then $z \in S$ such that z is the $i_k(x)^{\text{th}}$ element in C_k satisfy, by definition of $i_k(x)$, $z \leqslant x < y$, and therefore $z < y$ and $i_k(x) \leqslant i_k(y)$, therefore $\mathrm{lab}(x) \preccurlyeq \mathrm{lab}(y)$. Now, consider $x, y \in S$ such that $x || y$ and $L(x) < L(y)$. Suppose also, without loss of generality, that $x \in C_c$ for some integer c. By definition, x is the $i_c(x)^{\text{th}}$ smallest element in C_c. Note that, if $i_c(y) \geqslant i_c(x)$, then it implies that there exists $z \in C_c$ such that $x \leqslant z \leqslant y$, which would contradict our assumption that $x || y$. Therefore we have $i_c(y) < i_c(x)$ and $x < y$ which implies $\mathrm{lab}(x) || \mathrm{lab}(y)$. □

Claim 5.21 and Claim 5.22 combined imply that $Q_n(a, b)$ is strongly-universal for $\mathcal{Q}_n(a, b)$. The size constraint follows from the fact that the function $f(z_1 \dots, z_a) = \prod_{k \in [a]}(z_i + 1)$ is convex, and therefore, subject to $\sum_{k \in [a]} x_i = n$, it is maximise for $x_k = n/a$.

$$|S_n(m_1, \dots, m_n)| \leqslant na \prod_{k=1}^{a} (\ell_i + 1) \leqslant na \left(\frac{n}{a} + 1\right)^a.$$

□

5.5 Conclusion and perspectives

One of our main motivations in Section 5.3 was to obtain a finite version of the result of Huynh, Mohar, Šámal, Thomassen and Wood [93], stating that a countable graph containing all countable planar graphs as subgraph has an infinite clique minor. A natural question is whether there is a direct connection between the infinite and finite versions of the problem. While it does not seem to us that one result can be quickly deduced from the other, we note that our approach for producing K_t-minors out of short jumps can be used greedily in the infinite case to produce arbitrarily large clique minors, because in the infinite case all jumps can be considered as short. This alternative approach does not directly produce infinite clique minors as in [93] (additional compactness arguments are needed), but this highlights the fact that the finite version of the problem contains a number of challenges that do not

appear in the infinite version (such as boundary effects and the existence of long jumps), regardless of any quantitative aspects.

In terms of subgraph-universal K_t-minor-free graphs containing all K_s-minor-free graphs, we some additional result, not included in this paper, we settle the (approximate) order except for the case $s = 4$ and $t = 5, 6$. We show that K_5-minor-free subgraph-universal graphs of quasi-polynomial order can be obtained for the class of K_4-minor-free graphs and in fact those universal graphs have treewidth 3. This leaves the question of whether such a graph can be obtained of polynomial size.

Problem 5.23. *Does the class of K_4-minor-free graphs admit a subgraph-universal graph of polynomial order which is K_5-minor-free? If so, can it even be chosen of treewidth 3? If not, what if we allow it to be K_6-minor-free?*

The paper has been mostly dedicated to minor-closed classes, but interesting questions can be raised more generally for monotone or hereditary classes. The following problem was raised in [41], in the context of local certification in distributed computing.

Problem 5.24 ([41])**.** *For which graph H is it the case that the class of H-subgraph-free graphs has faithful subgraph-universal graphs of order $n \mapsto 2^{o(n^2)}$?*

In the case of universal posets, we have managed to improve the upper bound on the optimal size of a universal poset containing all posets in Section 5.4, but we are still far from completely solving the following question.

Problem 5.25. *What is the minimum size of a poset U_n containing all n-element posets as induced subposets?*

Chapter 6

Overall Conclusion and Perspectives

This chapter aims to reflect on the results established throughout this manuscript, and introduce some perspectives for future research.

6.1 Distance reconstruction

While we have precisely determined the query complexity for reconstructing several important graph classes—such as trees, k-chordal graphs, bounded treelength graphs, and Erdős–Rényi random graphs—the central open problem in distance reconstruction, originally posed by Mathieu and Zhou [140], remains unresolved.

Conjecture 2.5. *For any $\Delta \in \mathbb{N}$ there exists an algorithm A that reconstructs the class of all connected n-vertex graphs of maximum degree Δ using $O_\Delta(n \operatorname{polylog} n)$ queries w.h.p..*

Motivated by our new lower bound results, we propose a strengthened version of this conjecture, asking whether the $\operatorname{polylog} n$ factor can be improved to $O(\log n)$.

Conjecture 6.1. *For any $\Delta \in \mathbb{N}$ there exists an algorithm A which reconstructs the class of all n-vertex graphs of maximum degree Δ using $O_\Delta(n \log n)$ queries w.h.p..*

We also raised a question that is of both theoretical and practical significance. In Section 2.5, we analysed the query complexity of reconstructing $G(n,p)$ for small values of p. As in [141, 126], our result relies heavily on the expander properties of random graphs and random regular graphs, as well as the relatively tame distribution of degrees in these models. However, when aiming to reconstruct real-world networks such as the Internet, it is natural to consider graph classes with more heterogeneous degree distributions. In

particular, empirical studies show that many real-life networks follow a power-law degree distribution. This motivates the following open problem.

Problem 6.2. *Given a degree sequence D following a power-law distribution, what is the complexity of reconstructing a uniformly sampled graph with degree sequence D?*

6.2 Poset induced saturation

The main open questions in induced saturation revolve around deepening our understanding of the different possible behaviours and asymptotics of the function $\mathrm{sat}^*(n, P)$. It has been conjectured that this behaviour showcases a simple dichotomy.

Conjecture 6.3. *For any poset P and any $n \in \mathbb{N}$, either*

$$\mathrm{sat}^*(n, P) = O(1) \quad \textit{or} \quad \mathrm{sat}^*(n, P) = \Theta_P(n).$$

A first step towards this conjecture would be to prove the following weaker statement.

Conjecture 6.4. *There exists a constant $\alpha > 0$ such that for any poset P,*

$$\mathrm{sat}^*(n, P) = O_P(n^\alpha).$$

If we believe Conjecture 6.3 to be true, then a natural follow-up question would be to characterise the posets P such that $\mathrm{sat}^*(n, P) = O_P(1)$. Almost nothing is known about the structure of this family of posets. In particular, it is still unknown whether identifying those posets is even computable.

Problem 6.5. *What is the complexity class of the following algorithmic problem:*
Input: *A poset P*
Output: YES *if* $\mathrm{sat}^*(n, P) = O(1)$ *and* NO *otherwise.*

6.3 Spread embeddings

Kelly, Müyesser and Pokrovskiy suggested in [115], that *every* Dirac-type result might admit a spread version, regardless of the specific structure being embedded.

Problem 6.6. *For which family of graphs $\mathcal{H}$ and which value of $\delta \in [0, 1]$, is it true that for any n-vertex graph G of min degree δn there is an n-vertex graph $H \in \mathcal{H}$ such that there is a $O(1/n)$-spread embedding of H in G?*

A key reason this generalisation might be possible lies in the nature of Dirac-type theorems: their host graphs have linear minimum degree. Consequently, Chernoff bounds imply that almost all induced subgraphs of constant size ($O(1)$) in such dense graphs inherit a comparable (relative) minimum degree. When the target graph exhibits some recursive structure, this can be exploited in the design of a random embedding with optimal spread. The general approach would involve partitioning the target graph into $O(1)$-sized components—such as decomposing a Hamilton cycle into several subpaths, or trees into subtrees, as was done in Chapter 4. A natural step towards this challenging goal would be to generalise the bandwidth theorem to spread embeddings.

Conjecture 6.7. *For any $\alpha > 0$ and $r \geqslant 1$, there exists $\beta > 0$, such that given a n-vertex graph G of minimum degree $(\frac{r-1}{r} + \alpha)n$ and a n-vertex graph H of chromatic number r and bandwidth βn, H admits a $O(1/n)$-spread embedding in G.*

6.4 Faithful universal graphs and posets

Faithful universal structures have been thoroughly investigated in the setting of infinite graphs, yet many intriguing open problems remain in the finite case. For example, the following problem was raised in [41] in the context of local certification in distributed computing:

Problem 6.8 ([41])**.** *For which graphs H does the class of H-subgraph-free graphs admit faithful subgraph-universal graphs of order $n \mapsto 2^{o(n^2)}$?*

In [22], we also study faithful and near-faithful universal graphs for classes with bounded parameters (treewidth, pathwidth, treedepth, and maximum degree). Our results are summarised in Fig. 5.1 and Fig. 5.2. In particular, we constructed a polynomial-size induced-universal grap h U for the class of graphs with treewidth t, ensuring that $\mathbf{tw}(U) \leqslant 3t + 1$. This means that the increase in treewidth from the class to the induced-universal graph is bounded by a constant factor. In contrast, for pathwidth, Fig. 5.1 and Fig. 5.2 only guarantee an upper bound where the increase in pathwidth grows quadratically. We believe this discrepancy is not merely a limitation of our techniques, but may reflect a difference in behaviour between pathwidth and treewidth. This leads us to the following question:

Problem 6.9. *Is there a constant $C > 0$ such that for every $k \in \mathbb{N}$, the class $\mathcal{G}_{\mathrm{pw} \leqslant k}$ admits subgraph-universal graphs of pathwidth at most Ck and polynomial order $n^{O_k(1)}$?*

Bibliography

[1] Mikkel Abrahamsen, Stephen Alstrup, Jacob Holm, Mathias Bæk Tejs Knudsen, and Morten Stöckel. Near-optimal induced universal graphs for bounded degree graphs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 128:1–128:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[2] Mikkel Abrahamsen, Greg Bodwin, Eva Rotenberg, and Morten Stöckel. Graph reconstruction with a betweenness oracle. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, page 5. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2016.

[3] Wilhelm Ackermann. Die widerspruchsfreiheit der allgemeinen mengenlehre. *Mathematische Annalen*, 114(1):305–315, 1937.

[4] Ramtin Afshar and Michael T. Goodrich. Exact learning of multitrees and almost-trees using path queries. In Armando Castañeda and Francisco Rodríguez-Henríquez, editors, *LATIN 2022: Theoretical Informatics*, pages 293–311, 2022.

[5] Ramtin Afshar, Michael T. Goodrich, Pedro Matias, and Martha C. Osegueda. Reconstructing biological and digital phylogenetic trees in parallel. In *28th Annual European Symposium on Algorithms (ESA 2020)*, pages 3:1–3:24, 2020.

[6] Martin Aigner. Lexicographic matching in boolean algebras. *Journal of Combinatorial Theory, Series B*, 14:187–194, 1973.

[7] Peter Allen, Julia Böttcher, Jan Corsten, Ewan Davies, Matthew Jenssen, Patrick Morris, Barnaby Roberts, and Jozef Skokan. A robust Corrádi–Hajnal theorem. *Random Structures & Algorithms*, 65(1):61–130, 2024.

[8] Noga Alon. Asymptotically optimal induced universal graphs. *Geometric and Functional Analysis*, 27(1):1–32, February 2017.

[9] Noga Alon and Michael Capalbo. Optimal universal graphs with deterministic embedding. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, page 373–378, USA, 2008. Society for Industrial and Applied Mathematics.

[10] Noga Alon, Michael Capalbo, Yoshiharu Kohayakawa, Vojtěch Rödl, Andrzej Ruciński, and Endre Szemerédi. Near-optimum universal graphs for graphs with bounded degrees. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 170–180. Springer, 2001.

[11] Noga Alon and Rajko Nenadov. Optimal induced universal graphs for bounded-degree graphs. *Math. Proc. Camb. Philos. Soc.*, 166(1):61–74, 2019.

[12] Stephen Alstrup, Søren Dahlgaard, and Mathias Bæk Tejs Knudsen. Optimal induced universal graphs and adjacency labeling for trees. *J. ACM*, 64(4):27:1–27:22, 2017.

[13] Stephen Alstrup, Haim Kaplan, Mikkel Thorup, and Uri Zwick. Adjacency labeling schemes and induced-universal graphs. *SIAM J. Discrete Math.*, 33(1):116–137, 2019.

[14] Michael Anastos and Debsoumya Chakraborti. Robust hamiltonicity in families of Dirac graphs. *arXiv preprint arXiv:2309.12607*, 2023.

[15] Irina Đanković and Maria-Romina Ivan. Saturation for small antichains. *The Electronic Journal of Combinatorics*, pages P1–3, 2023.

[16] Maria Axenovich and Mónika Csikós. Induced saturation of graphs. *Discrete Mathematics*, 342(4):1195–1212, 2019.

[17] Laszlo Babai, Fan R. K. Chung, Paul Erdős, Ronald L. Graham, and Joel H. Spencer. On graphs which contain all sparse graphs. In *Theory and practice of combinatorics. A collection of articles honoring Anton Kotzig on the occasion of his sixtieth birthday*, pages 21–26. Elsevier, 1982.

[18] Béla Bajnok and Shahriar Shahriari. Long symmetric chains in the boolean lattice. *Journal of Combinatorial Theory, Series A*, 75(1):44–54, 1996.

[19] Paul Bastide. Distance reconstruction of sparse random graphs. *arXiv preprint arXiv:2407.17376*, 2024.

[20] Paul Bastide, Marthe Bonamy, Anthony Bonato, Pierre Charbit, Shahin Kamali, Théo Pierron, and Mikaël Rabie. Improved pyrotechnics: Closer to the burning number conjecture. *The Electronic Journal of Combinatorics*, 30(4), 2023.

[21] Paul Bastide, Linda Cook, Jeff Erickson, Carla Groenland, Marc van Kreveld, Isja Mannens, and Jordi L Vermeulen. Reconstructing graphs from connected triples. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 16–29. Springer, 2023.

[22] Paul Bastide, Louis Esperet, Carla Groenland, Claire Hilaire, Clément Rambaud, and Alexandra Wesolek. Faithful universal graphs for minor-closed classes. *arXiv preprint arXiv:2504.19582*, 2025.

[23] Paul Bastide and Pierre Fraigniaud. Brief annoucement: On extending brandt's speedup theorem from local to round-based full-information models. In *35th International Symposium on Distributed Computing (DISC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[24] Paul Bastide, George Giakkoupis, and Hayk Saribekyan. Self-stabilizing clock synchronization with 1-bit messages. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2154–2173. SIAM, 2021.

[25] Paul Bastide and Carla Groenland. Optimal distance query reconstruction for graphs without long induced cycles. *arXiv preprint arXiv:2306.05979*, 2023.

[26] Paul Bastide and Carla Groenland. Quasi-linear distance query reconstruction for graphs of bounded treelength. *19th International Symposium on Parameterized and Exact Computation (IPEC 2024)*, pages 20–1, 2024.

[27] Paul Bastide, Carla Groenland, Maria-Romina Ivan, and Tom Johnston. A polynomial upper bound for poset saturation. *European Journal of Combinatorics*, page 103970, 2024.

[28] Paul Bastide, Carla Groenland, Hugo Jacob, and Tom Johnston. Exact antichain saturation numbers via a generalisation of a result of lehman-ron. *Combinatorial Theory*, 4, 2024.

[29] Paul Bastide, Claire Hilaire, and Eileen Robinson. Path eccentricity of k-at-free graphs and application on graphs with the consecutive ones property. *arXiv preprint arXiv:2403.05360*, 2024.

[30] Paul Bastide, Clément Legrand-Duchesne, and Alp Müyesser. Random embeddings of bounded degree trees with optimal spread. *arXiv preprint arXiv:2409.06640*, 2024.

[31] Zuzana Beerliova, Felix Eberhard, Thomas Erlebach, Alexander Hall, Michael Hoffmann, Mat Mihal'ak, and L Shankar Ram. Network discovery and verification. *IEEE Journal on selected areas in communications*, 24(12):2168–2181, 2006.

[32] Sarah Behrens, Catherine Erbes, Michael Santana, Derrek Yager, and Elyse Yeager. Graphs with induced-saturation number zero. *The Electronic Journal of Combinatorics*, 23(1), 2016.

[33] Helena Bergold, Vesna Iršič, Robert Lauff, Joachim Orthaber, Manfred Scheucher, and Alexandra Wesolek. Subgraph-universal planar graphs for trees, 2024.

[34] Sandeep N. Bhatt, Fan R. K. Chung, Frank T. Leighton, and Arnold L. Rosenberg. Optimal simulations of tree machines (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 274–282. IEEE Computer Society, 1986.

[35] Sandeep N. Bhatt, Fan R. K. Chung, Frank T. Leighton, and Arnold L. Rosenberg. Universal graphs for bounded-degree trees and planar graphs. *SIAM Journal on Discrete Mathematics*, 2(2):145–155, 1989.

[36] Sandeep N. Bhatt and Charles E. Leiserson. How to assemble tree machines (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, page 77–84, New York, NY, USA, 1982. Association for Computing Machinery.

[37] Béla Bollobás. *Random graphs*. Springer, 1998.

[38] Marthe Bonamy, Louis Esperet, Carla Groenland, and Alex Scott. Optimal labelling schemes for adjacency, comparability, and reachability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1109–1117, 2021.

[39] Marthe Bonamy, Cyril Gavoille, and Michał Pilipczuk. Shorter labeling schemes for planar graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 446–462. SIAM, 2020.

[40] Julia Böttcher, Mathias Schacht, and Anusch Taraz. Proof of the bandwidth conjecture of Bollobás and Komlós. *Mathematische Annalen*, 343(1):175–205, 2009.

[41] Nicolas Bousquet, Louis Esperet, Laurent Feuilloley, and Sébastien Zeitoun. Renaming in distributed certification. *arXiv preprint arXiv:2409.15404*, 2024.

[42] Gerth Stølting Brodal, Rolf Fagerberg, Christian N. S. Pedersen, and Anna Östlin. The complexity of constructing evolutionary trees using experiments. *BRICS Report Series*, 8(1), 2001.

[43] Peter J Cameron. The random graph. *The Mathematics of Paul Erdös II*, pages 333–351, 1997.

[44] Peter J Cameron. The random graph revisited. In *European Congress of Mathematics: Barcelona, July 10–14, 2000, Volume I*, pages 267–274. Springer, 2001.

[45] Michael R. Capalbo. Small universal graphs for bounded-degree planar graphs. *Combinatorica*, 22(3):345–359, 2002.

[46] Amit Chakrabarti and Subhash Khot. Improved lower bounds on the randomized complexity of graph properties. In *28th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 285–296. Springer Verlag, 2001.

[47] Gregory Cherlin and Saharon Shelah. Universal graphs with a forbidden subtree. *Journal of Combinatorial Theory, Series B*, 97(3):293–333, 2007.

[48] Gregory Cherlin and Saharon Shelah. Universal graphs with a forbidden subgraph: block path solidity. *Combinatorica*, 36(3):249–264, 2016.

[49] Gregory Cherlin, Saharon Shelah, and Niandong Shi. Universal graphs with forbidden subgraphs and algebraic closure. *Advances in Applied Mathematics*, 22(4):454–491, 1999.

[50] Gregory Cherlin and Niandong Shi. Forbidden subgraphs and forbidden substructures. *The Journal of Symbolic Logic*, 66(3):1342–1352, 2001.

[51] Gregory Cherlin and Lasse Tallgren. Universal graphs with a forbidden near-path or 2-bouquet. *Journal of Graph Theory*, 56(1):41–63, 2007.

[52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[53] Fan R. K. Chung. Separator theorems and their applications. In *Paths, flows, and VLSI-layout, Proc. Meet., Bonn/Ger. 1988, Algorithms Comb. 9, 17-34 (1990)*, pages 17–34, 1990.

[54] Fan R. K. Chung. Universal graphs and induced-universal graphs. *Journal of Graph Theory*, 14(4):443–454, 1990.

[55] Fan R. K. Chung and Ronald L. Graham. On universal graphs for spanning trees. *Journal of the London Mathematical Society*, s2-27(2):203–211, 1983.

[56] Fan R. K. Chung, Ronald L. Graham, and Don Coppersmith. On trees which contain all small trees. *The Theory of Applications of Graphs*, pages 265–272, 1981.

[57] Fan R. K. Chung, Arnold L. Rosenberg, and Lawrence Snyder. Perfect storage representations for families of data structures. *SIAM Journal on Algebraic Discrete Methods*, 4(4):548–565, 1983.

[58] Bill Cuckler and Jeff Kahn. Hamiltonian cycles in Dirac graphs. *Combinatorica*, 29(3):299–326, 2009.

[59] Reinhard Diestel. On universal graphs with forbidden topological subgraphs. *European Journal of Combinatorics*, 6(2):175–182, 1985.

[60] Reinhard Diestel, Rudolf Halin, and Walter Vogler. Some remarks on universal graphs. *Combinatorica*, 5:283–293, 1985.

[61] Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, pages 161–166, 1950.

[62] G. A. Dirac. Some theorems on abstract graphs. *Proc. London Math. Soc. (3)*, 2:69–81, 1952.

[63] Gabriel Andrew Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, 3(1):69–81, 1952.

[64] Yon Dourisboure. Compact routing schemes for generalised chordal graphs. *J. Graph Algorithms Appl.*, 9(2):277–297, 2005.

[65] Yon Dourisboure and Cyril Gavoille. Tree-decompositions with bags of small diameter. *Discrete Mathematics*, 307(16):2008–2029, 2007.

[66] Dwight Duffus, David Howard, and Imre Leader. The width of downsets. *European Journal of Combinatorics*, 79:46–59, 2019.

[67] Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). *Journal of the ACM*, 68(6):Article 42, 2021. arXiv:2003.04280.

[68] Paul Erdos, András Hajnal, and John W Moon. A problem in graph theory. *The American Mathematical Monthly*, 71(10):1107–1110, 1964.

[69] Paul Erdos and Alfréd Rényi. Asymmetric graphs. *Acta Math. Acad. Sci. Hungar*, 14(295-315):3, 1963.

[70] Louis Esperet, Gwenaël Joret, and Pat Morin. Sparse universal graphs for planarity. *J. Lond. Math. Soc., II. Ser.*, 108(4):1333–1357, 2023.

[71] Jill R Faudree, Ralph J Faudree, and John R Schmitt. A survey of minimum saturated graphs. *The Electronic Journal of Combinatorics*, 1000:DS19–Jul, 2011.

[72] Uriel Feige. On sums of independent random variables with unbounded variance, and estimating the average degree in a graph. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 594–603, 2004.

[73] Michael Ferrara, Bill Kay, Lucas Kramer, Ryan R. Martin, Benjamin Reiniger, Heather C. Smith, and Eric Sullivan. The saturation number of induced subposets of the boolean lattice. *Discrete Mathematics*, 340(10):2479–2487, 2017.

[74] Pierre Fraigniaud and Amos Korman. An optimal ancestry scheme and small universal posets. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 611–620, 2010.

[75] Roland Fraïssé. *Theory of relations*. North-Holland, 1953.

[76] Keith Frankston, Jeff Kahn, Bhargav Narayanan, and Jinyoung Park. Thresholds versus fractional expectation-thresholds. *Ann. of Math. (2)*, 194(2):475–495, 2021.

[77] Andrea Freschi, Simón Piga, Maryam Sharifzadeh, and Andrew Treglown. The induced saturation problem for posets. *Combinatorial Theory*, 3(3), 2023.

[78] Zoltan Füredi and Péter Komjáth. Nonexistence of universal graphs without some trees. *Combinatorica*, 17:163–171, 1997.

[79] Zoltan Füredi and Péter Komjáth. On the existence of countable universal graphs. *Journal of Graph Theory*, 25(1):53–58, 1997.

[80] Cyril Gavoille and Arnaud Labourel. Shorter implicit representation for planar graphs and bounded treewidth graphs. In *Algorithms – ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8–10, 2007, Proceedings*, pages 582–593, 2007.

[81] Paweł Gawrychowski and Wojciech Janczewski. Simpler adjacency labeling for planar graphs with B-trees. In Karl Bringmann and Timothy Chan, editors, *5th Symposium on Simplicity in Algorithms, SOSA@SODA 2022, Virtual Conference, January 10-11, 2022*, pages 24–36. SIAM, 2022.

[82] Dániel Gerbner, Balázs Keszegh, Nathan Lemons, Cory Palmer, Dömötör Pálvölgyi, and Balázs Patkós. Saturating sperner families. *Graphs and Combinatorics*, 29(5):1355–1364, 2013.

[83] M. K. Gol'dberg and É. M. Livshits. On minimal universal trees. *Mathematical notes of the Academy of Sciences of the USSR*, 4:713–717, 1968.

[84] Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Structures & Algorithms*, 32(4):473–493, 2008.

[85] Mika Göös and T.S. Jayram. A composition theorem for conical juntas. In *31st Conference on Computational Complexity (CCC 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[86] Curtis Greene and Daniel J. Kleitman. Strong versions of sperner's theorem. *Journal of Combinatorial Theory, Series A*, 20(1):80–88, 1976.

[87] Pranshu Gupta, Fabian Hamann, Alp Müyesser, Olaf Parczyk, and Amedeo Sgueglia. A general approach to transversal versions of dirac-type theorems. *Bulletin of the London Mathematical Society*, 55(6):2817–2839, 2023.

[88] Frank Harary. A survey of the reconstruction conjecture. In *Graphs and Combinatorics: Proceedings of the Capital Conference on Graph Theory and Combinatorics at the George Washington University June 18–22, 1973*, pages 18–28. Springer, 2006.

[89] Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. In *63rd IEEE Symposium on Foundations of Computer Science (FOCS 2022)*, pages 1134–1137, 2022.

[90] Jotun Hein. An optimal algorithm to reconstruct trees from additive distance data. *Bulletin of mathematical biology*, 51(5):597–603, 1989.

[91] C Ward Henson. A family of countable homogeneous graphs. *Pacific journal of mathematics*, 38(1):69–83, 1971.

[92] Jan Hubička and Jaroslav Nešetřil. Universal partial order represented by means of oriented trees and other simple graphs. *European Journal of Combinatorics*, 26(5):765–778, 2005.

[93] Tony Huynh, Bojan Mohar, Robert Šámal, Carsten Thomassen, and David R Wood. Universality in minor-closed graph classes. *arXiv preprint arXiv:2109.00327*, 2021.

[94] Maria-Romina Ivan. Saturation for the butterfly poset. *Mathematika*, 66(3):806–817, 2020.

[95] Maria-Romina Ivan. Minimal diamond-saturated families. *arXiv preprint arXiv:2110.01118*, 2021.

[96] Maria-Romina Ivan and Sean Jaffe. Gluing posets and the dichotomy of poset saturation numbers. *arXiv preprint arXiv:2503.12223*, 2025.

[97] M. Jagadish and Anindya Sen. Learning a bounded-degree tree using separator queries. In Sanjay Jain, Rémi Munos, Frank Stephan, and Thomas Zeugmann, editors, *Algorithmic Learning Theory (ALT)*, pages 188–202, 2013.

[98] John B Johnston. Universal infinite partially ordered sets. *Proceedings of the American Mathematical Society*, 7(3):507–514, 1956.

[99] Bjarni Jónsson. Universal relational systems. *Mathematica Scandinavica*, pages 193–208, 1956.

[100] Felix Joos, Richard Lang, and Nicolás Sanhueza-Matamala. Robust hamiltonicity. *arXiv preprint arXiv:2312.15262*, 2023.

[101] Felix Joos, Richard Lang, and Nicolás Sanhueza-Matamala. Robust hamiltonicity, 2023.

[102] Felix Joos and Jonathan Schrodt. Counting oriented trees in digraphs with large minimum semidegree. *Journal of Combinatorial Theory, Series B*, 168:236–270, 2024.

[103] Jeff Kahn and Gil Kalai. Thresholds and expectation thresholds. *Combin. Probab. Comput.*, 16(3):495–502, 2007.

[104] Jeff Kahn, Michael Saks, and Dean Sturtevant. A topological approach to evasiveness. *Combinatorica*, 4:297–306, 1984.

[105] Dong Yeap Kang, Tom Kelly, Daniela Kühn, Deryk Osthus, and Vincent Pfenninger. Perfect matchings in random sparsifications of dirac hypergraphs. *Combinatorica*, pages 1–34, 2024.

[106] Sampath Kannan, Eugene Lawler, and Tandy Warnow. Determining the evolutionary tree using experiments. *Journal of Algorithms*, 21(1):26–50, 1996.

[107] Sampath Kannan, Claire Mathieu, and Hang Zhou. Near-linear query complexity for graph inference. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 773–784, 2015.

[108] Sampath Kannan, Claire Mathieu, and Hang Zhou. Graph reconstruction and verification. *ACM Transactions on Algorithms (TALG)*, 14(4):1–30, 2018.

[109] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM J. Discrete Math.*, 5(4):596–603, 1992.

[110] Ming-Yang Kao, Andrzej Lingas, and Anna Östlin. Balanced randomized tree splitting with applications to evolutionary tree constructions. In Christoph Meinel and Sophie Tison, editors, *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 184–196, 1999.

[111] Amarja Kathapurkar and Richard Montgomery. Spanning trees in dense directed graphs. *Journal of Combinatorial Theory, Series B*, 156:223–249, 2022.

[112] Gyula OH Katona and Tamás Gy Tarján. Extremal problems with excluded subgraphs in the n-cube. In *Graph Theory: Proceedings of a Conference held in Poland, February 10–13, 1981*, pages 84–93. Springer, 2006.

[113] Paul J Kelly. A congruence theorem for trees. *Pacific. J. Math.*, 1957.

[114] Tom Kelly, Alp Müyesser, and Alexey Pokrovskiy. Optimal spread for spanning subgraphs of Dirac hypergraphs. *Journal of Combinatorial Theory, Series B*, 169:507–541, 2024.

[115] Tom Kelly, Alp Müyesser, and Alexey Pokrovskiy. Optimal spread for spanning subgraphs of dirac hypergraphs. *Journal of Combinatorial Theory, Series B*, 169:507–541, 2024.

[116] Balázs Keszegh, Nathan Lemons, Ryan R. Martin, Dömötör Pálvölgyi, and Balázs Patkós. Induced and non-induced poset saturation problems. *Journal of Combinatorial Theory, Series A*, 184:105497, 2021.

[117] Valerie King, Li Zhang, and Yunhong Zhou. On the complexity of distance-based evolutionary tree reconstruction. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 444–453, USA, 2003. Society for Industrial and Applied Mathematics.

[118] Péter Komjáth. Some remarks on universal graphs. *Discrete mathematics*, 199(1-3):259–265, 1999.

[119] Péter Komjáth, Alan H Mekler, and János Pach. Some universal graphs. *Israel Journal of Mathematics*, 64(2):158–168, 1988.

[120] Péter Komjáth and János Pach. Universal graphs without large bipartite subgraphs. *Mathematika*, 31(2):282–290, 1984.

[121] J. Komlós, G. N. Sárközy, and E. Szemerédi. Proof of a packing conjecture of Bollobás. *Combin. Probab. Comput.*, 4(3):241–255, 1995.

[122] János Komlós, Gábor N. Sárközy, and Endre Szemerédi. Spanning trees in dense graphs. *Combin. Probab. Comput.*, 10(5):397–416, 2001.

[123] Adrian Kosowski, Bi Li, Nicolas Nisse, and Karol Suchan. k-chordal graphs: From cops and robber to compact routing via treewidth. *Algorithmica*, 72(3):758–777, 2015.

[124] Alexandr V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4:307–316, 1984.

[125] Michael Krivelevich, Choongbum Lee, and Benny Sudakov. Robust Hamiltonicity of Dirac graphs. *Trans. Amer. Math. Soc.*, 366(6):3095–3130, 2014.

[126] Michael Krivelevich and Maksim Zhukovskii. Reconstructing random graphs from distance queries. *arXiv preprint arXiv:2404.18318*, 2024.

[127] Joseph B. Kruskal. The number of simplices in a complex. *Mathematical optimization techniques*, 10:251–278, 1963.

[128] Eric Lehman and Dana Ron. On disjoint chains of subsets. *Journal of Combinatorial Theory, Series A*, 94(2):399–404, 2001.

[129] Nikos Leonardos. An improved lower bound for the randomized decision tree complexity of recursive majority. In Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming*, pages 696–708, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[130] Dingyuan Liu. Induced saturation for complete bipartite posets. *arXiv preprint arXiv:2402.08651*, 2024.

[131] Xizhi Liu and Sayan Mukherjee. Tight query complexity bounds for learning graph partitions. In *Proceedings of 35th Conference on Learning Theory (COLT)*, pages 167–181, 2022.

[132] Mark J. Logan. Sperner theory in a difference of boolean lattices. *Discrete Mathematics*, 257:501–512, 2002.

[133] Quentin Lutz, Elie de Panafieu, Alex Scott, and Maya Stein. Active clustering for labeling training data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 8469–8480, 2021.

[134] Frédéric Magniez, Ashwin Nayak, Miklos Santha, Jonah Sherman, Gábor Tardos, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. *Random Structures & Algorithms*, 48(3):612–638, 2016.

[135] W. Mantel. Problem 28, solution by H. Gouwentak, W. Mantel, J. Teixeira de Mattes, F. Schuh, and W.A. Wytho, 10. *Wiskundige Opgaven*, pages 60–61, 1907.

[136] Ryan R. Martin, Heather C. Smith, and Shanise Walker. Improved bounds for induced poset saturation. *The Electronic Journal of Combinatorics*, 27(2):P2.31, 2020.

[137] Ryan R Martin and Jason J Smith. Induced saturation number. *Discrete Mathematics*, 312(21):3096–3106, 2012.

[138] Ryan R Martin and Nick Veldt. Induced saturation of the poset 2c_2. *arXiv preprint arXiv:2408.14648*, 2024.

[139] Ryan R Martin and Nick Veldt. Saturation of k-chains in the boolean lattice. *arXiv preprint arXiv:2402.14113*, 2024.

[140] Claire Mathieu and Hang Zhou. Graph reconstruction via distance oracles. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 733–744, 2013.

[141] Claire Mathieu and Hang Zhou. A simple algorithm for graph reconstruction. *Random Structures & Algorithms*, pages 1–21, 2023.

[142] Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[143] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[144] Richard Montgomery. Spanning trees in random graphs. *Advances in Mathematics*, 356, 2019.

[145] Richard Montgomery, Alp Müyesser, and Yani Pehova. Transversal factors and spanning trees. *Adv. Comb.*, 2022.

[146] Natasha Morrison, Jonathan A. Noel, and Alex Scott. On saturated k-sperner systems. *The Electronic Journal of Combinatorics*, 21(3):P3.22, 2014.

[147] John H. Muller. *Local Structure in Graph Classes*. PhD thesis, School of Information and Computer Science, March 1988.

[148] János Pach. A problem of ulam on planar graphs. *Eur. J. Comb.*, 2(4):357–361, 1981.

[149] Jinyoung Park and Huy Pham. A proof of the kahn–kalai conjecture. *Journal of the American Mathematical Society*, 37(1):235–243, 2024.

[150] Jinyoung Park and Huy Pham. A proof of the Kahn–Kalai conjecture. *Journal of the American Mathematical Society*, 37(1):235–243, 2024.

[151] Matías Pavez-Signé, Nicolás Sanhueza-Matamala, and Maya Stein. Dirac-type conditions for spanning bounded-degree hypertrees. *Journal of Combinatorial Theory, Series B*, 165:97–141, 2024.

[152] Yanitsa Pehova and Kalina Petrova. Embedding loose spanning trees in 3-uniform hypergraphs. *Journal of Combinatorial Theory, Series B*, 168:47–67, 2024.

[153] Huy Tuan Pham, Ashwin Sah, Mehtaab Sawhney, and Michael Simkin. A toolkit for robust thresholds. *arXiv preprint arXiv:2210.03064*, 2022.

[154] Huy Tuan Pham, Ashwin Sah, Mehtaab Sawhney, and Michael Simkin. A toolkit for robust thresholds. arXiv:2210.03064, 2022.

[155] Richard Rado. Universal graphs and universal functions. *Acta Arithmetica*, 9:331–340, 1964.

[156] Lev Reyzin and Nikhil Srivastava. Learning and verifying graphs using queries with a focus on edge counting. In *Algorithmic Learning Theory: 18th International Conference, ALT 2007, Sendai, Japan, October 1-4, 2007. Proceedings 18*, pages 285–297. Springer, 2007.

[157] Ronald Rivest and Jean Vuillemin. A generalization and proof of the aanderaa-rosenberg conjecture. In *Proceedings of the seventh annual ACM Symposium on Theory of Computing (STOC)*, pages 6–11, 1975.

[158] Neil Robertson and Paul Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

[159] Neil Robertson and Paul Seymour. Graph minors xxiii. nash-williams' immersion conjecture. *Journal of Combinatorial Theory, Series B*, 100(2):181–205, 2010.

[160] Neil Robertson and Paul D Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.

[161] Guozhen Rong, Wenjun Li, Yongjie Yang, and Jianxin Wang. Reconstruction and verification of chordal graphs with a distance oracle. *Theoretical Computer Science*, 859:48–56, 2021.

[162] Guozhen Rong, Yongjie Yang, Wenjun Li, and Jianxin Wang. A divide-and-conquer approach for reconstruction of $C_{\geqslant 5}$-free graphs via betweenness queries. *Theoretical Computer Science*, 917:1–11, 2022.

[163] Arnold L Rosenberg. On the time required to recognize properties of graphs: A problem. *ACM SIGACT News*, 5(4):15–16, 1973.

[164] Jishnu Roychoudhury and Jatin Yadav. Efficient algorithms for sorting in trees. *arXiv:2205.15912*, 2022.

[165] Gábor N. Sárközy, Stanley M. Selkow, and Endre Szemerédi. On the number of Hamiltonian cycles in Dirac graphs. *Discrete Math.*, 265(1-3):237–250, 2003.

[166] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory. Series A*, 13:145–147, 1972.

[167] Norbert W. Sauer. On the density of families of sets. *J. Comb. Theory, Ser. A*, 13:145–147, 1972.

[168] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[169] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pac. J. Math.*, 41:247–261, 1972.

[170] Daniel Soudry, Suraj Keshri, Patrick Stinson, Min-hwan Oh, Garud Iyengar, and Liam Paninski. Efficient" shotgun" inference of neural connectivity from highly subsampled activity data. *PLoS computational biology*, 11(10):e1004464, 2015.

[171] Jeremy P. Spinrad. *Efficient graph representations*, volume 19 of *Fields Institute monographs*. American Mathematical Society, 2003.

[172] Clara Stegehuis and Lotte Weedage. Reconstruction of geometric random graphs with the simple algorithm. *arXiv preprint arXiv:2407.18591*, 2024.

[173] Michael Stiebitz, Thomas Schweser, and Bjarne Toft. *Brooks' Theorem: Graph Coloring and Critical Graphs*. Springer Nature, 2024.

[174] Benny Sudakov. Robustness of graph properties. In *Surveys in combinatorics 2017*, volume 440 of *London Math. Soc. Lecture Note Ser.*, pages 372–408. Cambridge Univ. Press, Cambridge, 2017.

[175] Michel Talagrand. Are many small sets explicitly small? In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 13–36, 2010.

[176] Andrew Thomason. An extremal function for contractions of graphs. *Math. Proc. Camb. Philos. Soc.*, 95:261–265, 1984.

[177] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 1–10, 2001.

[178] Paul Turán. On an external problem in graph theory. *Mat. Fiz. Lapok*, 48:436–452, 1941.

[179] Stanislaw M Ulam. A collection of mathematical problems. *Wiley*, 1960.

[180] Leslie G. Valiant. Universality considerations in VLSI circuits. *IEEE Transactions on Computers*, 30(2):135–140, February 1981.

[181] Zhaosen Wang and Jean Honorio. Reconstructing a bounded-degree directed tree using path queries. In *57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 506–513, 2019.

[182] Michael Waterman, Temple Smith, Mona Singh, and William A Beyer. Additive evolutionary trees. *Journal of Theoretical Biology*, 64(2):199–213, 1977.

[183] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 222–227. IEEE Computer Society, 1977.